

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application:

2 0 0 4 年 3 月 1 8 日

出 願 番 号
Application Number:

特願 2 0 0 4 - 0 7 8 9 7 9

[ST. 10/C]:

[J P 2 0 0 4 - 0 7 8 9 7 9]

願 人
Applicant(s):

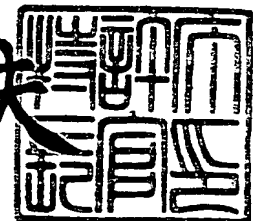
株式会社リコー

CERTIFIED COPY OF
PRIORITY DOCUMENT

2 0 0 4 年 4 月 8 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願
【整理番号】 0309866
【提出日】 平成16年 3月18日
【あて先】 特許庁長官 今井 康夫 殿
【国際特許分類】 G03G 21/00 370
【発明者】
 【住所又は居所】 東京都大田区中馬込 1 丁目 3 番 6 号 株式会社リコー内
 【氏名】 安藤 光男
【特許出願人】
 【識別番号】 000006747
 【氏名又は名称】 株式会社リコー
【代理人】
 【識別番号】 100070150
 【弁理士】
 【氏名又は名称】 伊東 忠彦
【先の出願に基づく優先権主張】
 【出願番号】 特願2003- 82908
 【出願日】 平成15年 3月25日
【手数料の表示】
 【予納台帳番号】 002989
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9911477

【書類名】 特許請求の範囲**【請求項 1】**

画像形成処理におけるハードウェア資源の制御に関するサービスを複数のアプリケーションに共通に提供する制御部を備えることにより、複数のアプリケーションを搭載可能に構成された画像形成装置であって、

前記アプリケーションとしての仮想マシンと、

前記仮想マシン上で動作するアプリケーションと、

前記仮想マシン上で動作するアプリケーションを管理するためのアプリケーション管理部とを備えたことを特徴とする画像形成装置。

【請求項 2】

前記仮想マシン上で動作するアプリケーションは、前記画像形成装置を制御するためのクラスを用いて画像形成装置を操作する請求項 1 に記載の画像形成装置。

【請求項 3】

前記クラスとして、前記画像形成装置の操作パネルへの情報の出力及び情報の入力を行うための操作パネルクラスを有する請求項 2 に記載の画像形成装置。

【請求項 4】

前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションの起動を含む実行管理機能を有する請求項 1 に記載の画像形成装置。

【請求項 5】

前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションがインストールされていない場合に、アプリケーションのダウンロードを行う請求項 4 に記載の画像形成装置。

【請求項 6】

前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルを参照し、当該アプリケーション情報ファイルの情報に基づき前記アプリケーションの更新を行う請求項 1 に記載の画像形成装置。

【請求項 7】

前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルに記述されたバージョンと、更新用アプリケーションに付随するアプリケーション情報ファイルに記述されたバージョンとを比較し、更新用アプリケーションに付随するアプリケーション情報ファイルに記述されたバージョンのほうが新しい場合に、前記アプリケーションの更新を行う請求項 6 に記載の画像形成装置。

【請求項 8】

前記アプリケーション管理部は、メモ리카ード又は Web サイトからアプリケーションをダウンロードする請求項 5 に記載の画像形成装置。

【請求項 9】

前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルに記述された場所からアプリケーションをダウンロードする請求項 6 に記載の画像形成装置。

【請求項 10】

前記アプリケーション管理部は、前記アプリケーションをダウンロードする前に、そのアプリケーションが前記画像形成装置で実行可能か否かの確認を行う請求項 5 又は 6 に記載の画像形成装置。

【請求項 11】

前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルに記述された情報に基づき前記確認を行う請求項 10 に記載の画像形成装置。

【請求項 12】

前記確認を行う際に、前記アプリケーションが必要とする前記画像形成装置のリソースを含む事項の確認を行う請求項 10 に記載の画像形成装置。

【請求項 13】

前記仮想マシン上で動作するアプリケーションとして、ネットワークを介して前記画像形成装置に接続された端末装置からの操作要求を受けて前記画像形成装置を操作するサーバアプリケーションを有する請求項 1 に記載の画像形成装置。

【請求項 1 4】

前記アプリケーション管理部は、前記端末装置と通信することにより処理を行う請求項 1 3 に記載の画像形成装置。

【請求項 1 5】

複数のアプリケーションを搭載可能に構成された画像形成装置を操作する端末装置であって、

仮想マシンと、

前記仮想マシン上で動作し、前記画像形成装置を操作するアプリケーションと、

前記仮想マシン上で動作するアプリケーションを管理するためのアプリケーション管理部とを備えたことを特徴とする端末装置。

【請求項 1 6】

前記仮想マシン上で動作するアプリケーションは、ネットワークを介して前記端末装置に接続された前記画像形成装置を制御するためのクラスを用いて前記画像形成装置を操作する請求項 1 5 に記載の端末装置。

【請求項 1 7】

前記画像形成装置の操作を J a v a (登録商標) に基づく分散処理技術を用いて行う請求項 1 5 又は 1 6 に記載の端末装置。

【請求項 1 8】

前記クラスは、前記画像形成装置の操作パネルへの表示画面をエミュレートした画面を前記端末装置に表示する機能を含む請求項 1 6 に記載の端末装置。

【請求項 1 9】

前記画像形成装置のハードディスクと前記端末装置のハードディスクのうちいずれのハードディスクにアクセスするかを選択させる機能を有する請求項 1 5 に記載の端末装置。

【請求項 2 0】

前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションの起動を含む実行管理機能を有する請求項 1 5 に記載の端末装置。

【請求項 2 1】

前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションがインストールされていない場合に、アプリケーションのダウンロードを前記画像形成装置と通信することにより行う請求項 2 0 に記載の端末装置。

【請求項 2 2】

前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルを参照し、当該アプリケーション情報ファイルの情報に基づき前記アプリケーションの更新を行う請求項 1 5 に記載の端末装置。

【請求項 2 3】

前記ダウンロードは、前記画像形成装置に挿入するメモリカード又は W e b サイトから行う請求項 2 1 に記載の端末装置。

【請求項 2 4】

前記アプリケーション管理部は、前記メモリカードに格納されたファイル情報を前記画像形成装置から受信し、表示をする請求項 2 3 に記載の端末装置。

【請求項 2 5】

前記アプリケーション管理部は、前記端末装置に入力された U R L 情報を前記画像形成装置に送信し、前記画像形成装置が該 U R L 情報に基づき前記仮想マシン上で動作するアプリケーションを W e b サイトからダウンロードする請求項 2 3 に記載の端末装置。

【請求項 2 6】

複数のアプリケーションを搭載可能に構成された画像形成装置を、ネットワークを介して接続された端末装置上で実行するアプリケーションにより操作する方法であって、

前記画像形成装置に、前記アプリケーションとしての仮想マシンと、該仮想マシン上で動作するサーバーアプリケーションとを備え、

前記端末装置に、仮想マシンと、該仮想マシン上で動作するアプリケーションとを備え、

前記端末装置のアプリケーションによる操作要求を前記サーバーアプリケーションが受信して、該サーバーアプリケーションが前記画像形成装置を操作することを特徴とする方法。

【請求項 2 7】

前記端末装置からの前記画像形成装置の操作は、J a v a（登録商標）に基づく分散処理技術を用いて行う請求項 2 6 に記載の方法。

【請求項 2 8】

前記画像形成装置にアプリケーションを管理するためのアプリケーション管理部を備え、前記端末装置に、該アプリケーション管理部と通信することによりアプリケーションを管理する遠隔アプリケーション管理部を備える請求項 2 6 に記載の方法。

【請求項 2 9】

画像形成処理におけるハードウェア資源の制御に関するサービスを複数のアプリケーションに共通に提供する制御部を備えることにより、複数のアプリケーションを搭載可能に構成された画像形成装置に、

前記アプリケーションとしての仮想マシンの機能と、

前記仮想マシン上で動作するアプリケーションを管理するためのアプリケーション管理部の機能とを実現させるためのプログラム。

【請求項 3 0】

前記仮想マシン上で動作するアプリケーションにより用いられる、前記画像形成装置を制御するためのクラスを含む請求項 2 9 に記載のプログラム。

【請求項 3 1】

前記クラスは、前記画像形成装置の操作パネルへの情報の出力及び情報の入力を行うための操作パネルクラスを含む請求項 3 0 に記載のプログラム。

【請求項 3 2】

前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションの起動を含む実行管理機能を有する請求項 2 9 に記載のプログラム。

【請求項 3 3】

前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションがインストールされていない場合に、アプリケーションのダウンロードを行う機能を有する請求項 3 2 に記載のプログラム。

【請求項 3 4】

前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルを参照し、当該アプリケーション情報ファイルの情報に基づき前記アプリケーションの更新を行う機能を有する請求項 2 9 に記載のプログラム。

【請求項 3 5】

前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルに記述されたバージョンと、更新用アプリケーションに付随するアプリケーション情報ファイルに記述されたバージョンとを比較し、更新用アプリケーションに付随するアプリケーション情報ファイルに記述されたバージョンのほうが新しい場合に、前記アプリケーションの更新を行う機能を有する請求項 3 4 に記載のプログラム。

【請求項 3 6】

前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルに記述された場所からアプリケーションをダウンロードする機能を有する請求項 3 4 に記載のプログラム。

【請求項 3 7】

前記アプリケーション管理部は、前記アプリケーションをダウンロードする前に、その

アプリケーションが前記画像形成装置で実行可能か否かの確認を行う機能を有する請求項 3 3 又は 3 4 に記載のプログラム。

【請求項 3 8】

前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルに記述された情報に基づき前記確認を行う機能を有する請求項 3 7 に記載のプログラム。

【請求項 3 9】

前記アプリケーション管理部は、前記確認を行う際に、前記アプリケーションが必要とする前記画像形成装置のリソースを含む事項の確認を行う機能を有する請求項 3 7 に記載のプログラム。

【請求項 4 0】

前記仮想マシン上で動作するアプリケーションは、ネットワークを介して前記画像形成装置に接続された端末装置からの操作要求を受けて前記画像形成装置を操作するサーバーアプリケーションを含む請求項 2 9 に記載のプログラム。

【請求項 4 1】

複数のアプリケーションを搭載可能に構成された画像形成装置を操作する端末装置に、仮想マシンの機能と、
前記仮想マシン上で動作し、前記画像形成装置を操作するためのアプリケーションを管理するためのアプリケーション管理部の機能とを実現させるためのプログラム。

【請求項 4 2】

前記仮想マシン上で動作するアプリケーションにより用いられる、前記画像形成装置を制御するためのクラスを含む請求項 4 1 に記載のプログラム。

【請求項 4 3】

前記画像形成装置の操作を J a v a（登録商標）に基づく分散処理技術を用いて行う請求項 4 1 又は 4 2 に記載のプログラム。

【請求項 4 4】

前記クラスは、前記画像形成装置の操作パネルへの表示画面をエミュレートした画面を前記端末装置に表示する機能を含む請求項 4 2 に記載のプログラム。

【請求項 4 5】

前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションの起動を含む実行管理機能を有する請求項 4 1 に記載のプログラム。

【請求項 4 6】

前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションがインストールされていない場合に、アプリケーションのダウンロードを前記画像形成装置と通信することにより行う機能を有する請求項 4 5 に記載のプログラム。

【請求項 4 7】

前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルを参照し、当該アプリケーション情報ファイルの情報に基づき前記アプリケーションの更新を行う機能を有する請求項 4 1 に記載のプログラム。

【請求項 4 8】

請求項 2 9 ないし 4 7 のうちいずれか 1 項に記載のプログラムを記録したコンピュータ読み取り可能な記録媒体。

【書類名】 明細書

【発明の名称】 画像形成装置及び画像形成装置を遠隔のアプリケーションにより操作する方法

【技術分野】**【0 0 0 1】**

この発明は、J a v a（登録商標）アプリケーションにより画像形成装置を操作する技術に関する。

【背景技術】**【0 0 0 2】**

近年では、プリンタ、コピー、ファクシミリ、スキャナなどの各装置の機能を1つの筐体内に収納した画像形成装置（以下、「複合機」という。）が知られている。この複合機は、1つの筐体内に表示部、印刷部および撮像部などを設けるとともに、プリンタ、コピーおよびファクシミリ装置にそれぞれ対応した3種類のソフトウェアを設け、これらのソフトウェアを切り替えることによって、当該装置をプリンタ、コピー、スキャナまたはファクシミリ装置として動作させるものである。

【0 0 0 3】

このようなソフトウェアの切り替えによって実現されている複合機は、プリンタ、コピー、スキャナ及びファクシミリ装置に対応するアプリケーション及びシステムソフトウェアをそれぞれ別個に設けているため、各アプリケーション及びシステムソフトウェアの開発に多大な時間を要する。この問題を解決するために、アプリケーションの少なくとも2つが共通に必要とするハードウェア資源の管理、実行制御並びに画像形成処理を行うプラットフォームを備えた構成の複合機が開発されている。このような複合機では、アプリケーションと、ハードウェア資源にアクセスするような開発が難しい処理を行うコントロールサービスとを別個に設けているため、複合機の出荷後にユーザもしくは第三者であるサードベンダが画像形成処理などにかかるアプリケーションとして新規なアプリケーションを開発して複合機に搭載可能な構成となっている。

【0 0 0 4】

搭載するアプリケーションとして種々のものが開発されているが、複数のプラットフォームに対応し、ネットワークに高度に対応するという観点から、上記の複合機に搭載するアプリケーションとしてJ a v a（登録商標）アプリケーションを用いることが検討されている。J a v a（登録商標）アプリケーションに関する従来技術として例えば特許文献1に記載された技術がある。

【特許文献1】 特開 2 0 0 2 - 2 8 7 9 9 0 号公報

【発明の開示】**【発明が解決しようとする課題】****【0 0 0 5】**

J a v a（登録商標）アプリケーションを上記の複合機のアプリケーションとして実行するためには、J a v a（登録商標）の実行環境としてJ a v a（登録商標）アプリケーションの管理を行うモジュールが必要である。

【0 0 0 6】

また、J a v a（登録商標）アプリケーションを用いて複合機のアプリケーションを開発する場合、P Cなどの上で開発を行った後に、実際に複合機上で動作するか否かの評価を行う必要があるが、評価の度に複合機にアプリケーションのインストールをして実行することは煩雑である。従って、開発したアプリケーションをP Cなどの開発環境から直接複合機で実行できるような環境が求められている。また、遠隔でのアプリケーションの実行により複合機を操作することができれば、J a v a（登録商標）アプリケーションを使用するユーザーの利便性が向上する。

【0 0 0 7】

本発明は上記の点に鑑みてなされたものであり、J a v a（登録商標）アプリケーションを複合機上で容易に実行できる環境、及びP Cなどの端末装置で実行するJ a v a（登

録商標) アプリケーションにより複合機を操作できる環境を提供することを目的とする。

【課題を解決するための手段】

【0008】

上記の課題は下記の発明により解決される。

【0009】

請求項1に記載の発明は、画像形成処理におけるハードウェア資源の制御に関するサービスを複数のアプリケーションに共通に提供する制御部を備えることにより、複数のアプリケーションを搭載可能に構成された画像形成装置に、前記アプリケーションとしての仮想マシンと、前記仮想マシン上で動作するアプリケーションと、前記仮想マシン上で動作するアプリケーションを管理するためのアプリケーション管理部とを備えるものである。

【0010】

このように、画像形成装置のアプリケーション層に仮想マシンを搭載したので、仮想マシン上で動作するアプリケーションであるJava(登録商標)アプリケーションを実行でき、また、アプリケーション管理部を設けたのでJava(登録商標)アプリケーションの追加、削除、実行管理などを容易に行うことができることから、本発明により複合機上でJava(登録商標)アプリケーションを容易に実行できるようになる。

【0011】

請求項2に記載の発明は、請求項1の記載において、前記仮想マシン上で動作するアプリケーションは、前記画像形成装置を制御するためのクラスを用いて画像形成装置を操作するものである。

【0012】

請求項3に記載の発明は、請求項2の記載において、前記クラスとして、前記画像形成装置の操作パネルへの情報の出力及び情報の入力を行うための操作パネルクラスを有するものである。このようにクラスを用いることにより、容易に画像形成装置のためのアプリケーションを開発できる。

請求項4に記載の発明は、請求項1の記載において、前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションの起動を含む実行管理機能を有するものである。また、請求項5の発明は、前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションがインストールされていない場合に、アプリケーションのダウンロードを行うこととするものである。アプリケーション管理部をこのような構成とすることにより、アプリケーションの起動やインストールを容易に行うことができる。

【0013】

また、請求項6に記載の発明は、請求項1の記載において、前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルを参照し、当該アプリケーション情報ファイルの情報に基づき前記アプリケーションの更新を行うものである。本発明によれば、アプリケーション情報ファイルの設定に基づきアプリケーションの更新を行うことができるので、例えばアプリケーションを自動的に更新できる。

【0014】

請求項7に記載の発明は、請求項6の記載において、前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルに記述されたバージョンと、更新用アプリケーションに付随するアプリケーション情報ファイルに記述されたバージョンとを比較し、更新用アプリケーションに付随するアプリケーション情報ファイルに記述されたバージョンのほうが新しい場合に、前記アプリケーションの更新を行うものである。これにより、バージョンの古いアプリケーションを自動的に更新できるようになる。

【0015】

請求項8に記載の発明は、請求項5の記載において、前記アプリケーション管理部は、メモリカード又はWebサイトからアプリケーションをダウンロードするものであり、請求項9に記載の発明は、請求項6の記載において、前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルに記述された場所からアプリ

ケーションをダウンロードするものである。

【0016】

また、請求項10に記載の発明は、前記アプリケーションをダウンロードする前に、そのアプリケーションが前記画像形成装置で実行可能か否かの確認を行うこととするものであり、その確認は、請求項11のように、前記アプリケーションに付随するアプリケーション情報ファイルに記述された情報に基づき行うことができる。また、請求項12に記載されているように、前記確認を行う際に、前記アプリケーションが必要とする前記画像形成装置のリソースを含む事項の確認を行う。アプリケーション管理部がこのような機能を有することにより、容易かつ確実にインストール、及び更新を行うことが可能になる。

【0017】

請求項13に記載の発明は、請求項1の記載において、前記仮想マシン上で動作するアプリケーションとして、ネットワークを介して前記画像形成装置に接続された端末装置からの操作要求を受けて前記画像形成装置を操作するサーバーアプリケーションを有するものである。

【0018】

本発明によれば、端末装置で実行されるアプリケーションに基づく操作手順に従って画像形成装置を操作することが可能となる。請求項14に記載されているように、この場合、前記アプリケーション管理部は、前記端末装置と通信することにより処理を行う。

【0019】

また、請求項15に記載の発明は、複数のアプリケーションを搭載可能に構成された画像形成装置を操作する端末装置であって、仮想マシンと、前記仮想マシン上で動作し、前記画像形成装置を操作するアプリケーションと、前記仮想マシン上で動作するアプリケーションを管理するためのアプリケーション管理部とを備えたものである。

【0020】

本発明によれば、端末装置で実行するアプリケーションにより上記の画像形成装置を操作することが可能となる。

【0021】

請求項16に記載の発明は、前記仮想マシン上で動作するアプリケーションは、ネットワークを介して前記端末装置に接続された前記画像形成装置を制御するためのクラスを用いて前記画像形成装置を操作するものである。請求項17に記載の発明は、前記画像形成装置の操作をJava（登録商標）に基づく分散処理技術を用いて行うこととしたものである。このように、ネットワークに高度に対応したJava（登録商標）の分散処理技術を用いることにより、ネットワーク上の任意の装置で実行したアプリケーションの手順に従って画像形成装置を操作することが可能となる。

【0022】

請求項18に記載の発明は、前記クラスは、前記画像形成装置の操作パネルへの表示画面をエミュレートした画面を前記端末装置に表示する機能を含むものである。これにより、端末装置上で、複合機を操作する場合と同様の操作を行うことが可能となる。

【0023】

請求項19に記載の発明は、請求項15の端末装置が前記画像形成装置のハードディスクと前記端末装置のハードディスクのうちいずれのハードディスクにアクセスするかを選択させる機能を有するものである。本発明により、アプリケーションの実行に際して端末装置から複合機のハードディスクに格納されたファイルを選択して、例えば印刷させることが可能となる。

【0024】

請求項20に記載の発明は、請求項15の記載において、前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションの起動を含む実行管理機能を有するものであり、また、請求項21に記載されているように、前記アプリケーション管理部は、前記仮想マシン上で動作するアプリケーションがインストールされていない場合に、アプリケーションのダウンロードを前記画像形成装置と通信することにより行う。また、請求

項 22 に記載されているように、前記アプリケーション管理部は、前記アプリケーションに付随するアプリケーション情報ファイルを参照し、当該アプリケーション情報ファイルの情報に基づき前記アプリケーションの更新を行う。更に、請求項 23 に記載されているように、前記ダウンロードは、前記画像形成装置に挿入するメモリカード又は Web サイトから行うことができる。

【0025】

請求項 24 に記載の発明は、請求項 23 の記載において、前記アプリケーション管理部は、前記メモリカードに格納されたファイル情報を前記画像形成装置から受信し、表示をすることとしたものである。また、請求項 25 に記載の発明は、請求項 23 の記載において、前記アプリケーション管理部は、前記端末装置に入力された URL 情報を前記画像形成装置に送信し、前記画像形成装置が該 URL 情報に基づき前記仮想マシン上で動作するアプリケーションを Web サイトからダウンロードするようにしたものである。このように、アプリケーション管理部が画像形成装置と通信しながら処理を行うので、画像形成装置が有しているアプリケーションの管理機能を端末装置に拡張できる。

【0026】

請求項 26 に記載の発明は、複数のアプリケーションを搭載可能に構成された画像形成装置を、ネットワークを介して接続された端末装置上で実行するアプリケーションにより操作する方法であって、前記画像形成装置に、前記アプリケーションとしての仮想マシンと、該仮想マシン上で動作するサーバーアプリケーションとを備え、前記端末装置に、仮想マシンと、該仮想マシン上で動作するアプリケーションとを備え、前記端末装置のアプリケーションによる操作要求を前記サーバーアプリケーションが受信して、該サーバーアプリケーションが前記画像形成装置を操作するものである。

【0027】

本発明によれば、複合機の機能を有さない PC などの端末でアプリケーションを実行することにより、画像形成装置を操作できる。

【0028】

請求項 27 に記載の発明は、請求項 26 に記載の発明において、前記端末装置からの前記画像形成装置の操作は、Java（登録商標）に基づく分散処理技術を用いて行うこととしたものである。また、請求項 28 に記載の発明は、請求項 26 に記載の発明において、前記画像形成装置にアプリケーションを管理するためのアプリケーション管理部を備え、前記端末装置に、該アプリケーション管理部と通信することによりアプリケーションを管理する遠隔アプリケーション管理部を備えるものである。このように、アプリケーション管理部と通信することによりアプリケーションを管理する遠隔アプリケーション管理部を備えることにより、画像形成装置のアプリケーション管理機能を端末装置に拡張することができる。

【0029】

請求項 29～40 に記載の発明は、上記画像形成装置の機能を実現するプログラムであり、請求項 41～47 に記載の発明は、上記端末装置の機能を実現するプログラムであり、請求項 48 に記載の発明は、上記プログラムを格納する記録媒体の発明である。

【発明の効果】

【0030】

上記のように本発明の Java（登録商標）実行環境を複合機に備えることにより、複合機上で容易に Java（登録商標）アプリの起動やインストールなどを行うことができるようになることから、Java（登録商標）アプリを容易に実行できる。また、本発明によれば、PC で Java（登録商標）アプリを実行して複合機を操作することが可能となる。

【発明を実施するための最良の形態】

【0031】

以下、図面を参照して本発明の実施の形態について説明する。第 1、第 2 の実施の形態として複合機における Java（登録商標）実行環境について説明し、第 3、第 4 の実施

の形態として、PCなどで開発したアプリケーションを、PCで実行させることにより複合機を操作する環境について説明する。まず、第1～第3の実施の形態で用いられる複合機について説明する。

【0032】

図1は、本発明の実施の形態における複合機の構成を示すブロック図である。図1に示すように、複合機100は、白黒レーザプリンタ(B&W LP)101と、カラーレーザプリンタ(Color LP)102と、スキャナ、ファクシミリ、ハードディスク、メモリ、ネットワークインタフェースなどのハードウェアリソース103を有する。また、各種コントロールサービスを有するプラットフォーム120と、アプリケーション(以下、アプリともいう)130とを備えている。

【0033】

プラットフォーム120は、アプリケーションからの処理要求を解釈してハードウェア資源の獲得要求を発生させるコントロールサービスと、一又は複数のハードウェア資源の管理を行い、コントロールサービスからの獲得要求を調停するシステムリソースマネージャ(ステップSRM)123と、汎用OS121とを有している。

【0034】

コントロールサービスは、複数のサービスモジュールから形成され、SCS(システムコントロールサービス)122と、ECS(エンジンコントロールサービス)124と、MCS(メモリコントロールサービス)125と、OCS(オペレーションパネルコントロールサービス)126と、FCS(ファックスコントロールサービス)127と、NCS(ネットワークコントロールサービス)128とから構成されている。なお、このプラットフォーム120は、あらかじめ定義された関数により前記アプリケーション130から処理要求を受信可能とするアプリケーションプログラムインタフェース(API)を有している。

【0035】

汎用OS121は、UNIX(登録商標)などの汎用オペレーティングシステムであり、プラットフォーム120並びにアプリケーション130の各ソフトウェアをそれぞれプロセスとして並列実行する。

【0036】

SRM123のプロセスは、SCS122とともにシステムの制御及びリソースの管理を行うものである。SRM123のプロセスは、スキャナ部やプリンタ部などのエンジン、メモリ、HDDファイル、ホストI/O(セントロI/F、ネットワークI/F、IEEE1394 I/F、RS232C I/Fなど)のハードウェア資源を利用する上位層からの要求にしたがって調停を行い、実行制御する。

【0037】

SCS122のプロセスは、アプリ管理、オペレーションパネル制御、システム画面表示、LED表示、リソース管理、割り込みアプリ制御などを行う。また、ECS124のプロセスは、白黒レーザプリンタ(B&W LP)101、カラーレーザプリンタ(Color LP)102、スキャナ、ファクシミリなどからなるハードウェアリソースのエンジンの制御を行う。

【0038】

MCS125のプロセスは、画像メモリの取得及び解放、ハードディスク装置(HDD)の利用、画像データの圧縮及び伸張などを行い、FCS127のプロセスは、システムコントローラの各アプリ層からPSTN/ISDN網を利用したファクシミリ送受信、BKM(バックアップSRAM)で管理されている各種ファクシミリデータの登録/引用、ファクシミリ読みとり、ファクシミリ受信印刷、融合送受信を行うためのAPIを提供する。

【0039】

NCS128のプロセスは、ネットワークI/Oを必要とするアプリケーションに対して共通に利用できるサービスを提供するためのプロセスであり、ネットワーク側から各プ

ロトコルによって受信したデータを各アプリケーションに振り分けたり、アプリケーションからデータをネットワーク側に送信する際の仲介を行う。具体的には、ftpd、httpd、lpd、snmpd、telnetd、smtpdなどのサーバデーモンや、同プロトコルのクライアント機能などを有している。

【0040】

OCS126のプロセスは、オペレータ（ユーザ）と本体制御間の情報伝達手段となるオペレーションパネル（操作パネルとも称する）の制御を行う。OCS126は、オペレーションパネルからのキー押下情報をキーイベントとして取得し、取得したキーに対応したキーイベント関数をSCS122に送信するOCSプロセスの部分と、アプリケーション130又はコントロールサービスからの要求によりオペレーションパネルに各種画面を描画出力する描画関数やその他オペレーションパネルに対する制御を行う関数などがあらかじめ登録されたOCSライブラリの部分とから構成される。このOCSライブラリは、アプリケーション130及びコントロールサービスの各モジュールにリンクされて実装されている。

【0041】

アプリケーション130は、ページ記述言語（PDL）、PCL及びポストスクリプト（PS）を有するプリンタ用のアプリケーションであるプリンタアプリ111と、コピー用アプリケーションであるコピーアプリ112と、ファクシミリ用アプリケーションであるファックスアプリ113と、スキャナ用アプリケーションであるスキャナアプリ114と、ネットワークファイル用アプリケーションであるネットファイルアプリ115とを有している。また、アプリケーションとして、Java（登録商標）実行環境116とJava（登録商標）アプリケーション117（以下、Java（登録商標）アプリと称する）を搭載している。図1では、Java（登録商標）アプリが複数搭載されている例を示している。

【0042】

アプリケーション130の各プロセス、コントロールサービスの各プロセスは、関数呼び出しとその戻り値送信及びメッセージの送受信によってプロセス間通信を行いながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成処理にかかるユーザサービスを実現している。

【0043】

このように、本実施の形態にかかる複合機100には、複数のアプリケーション及び複数のコントロールサービスが存在し、いずれもプロセスとして動作している。そして、これらの各プロセス内部には、一又は複数のスレッドが生成されて、スレッド単位の並列実行が行われる。そして、コントロールサービスがアプリケーションに対し共通サービスを提供しており、このため、これらの多数のプロセスが並列動作、及びスレッドの並列動作を行って互いにプロセス間通信を行って協調動作をしながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成処理にかかるユーザサービスを提供するようになっている。また、複合機100には、サードベンダなどの第三者がコントロールサービス層の上のアプリケーション層に新規アプリを開発して搭載することが可能となっている。

【0044】

図2は、複合機100のハードウェア構成図である。図2に示すように、複合機100は、コントローラ40と、オペレーションパネル60と、ファックスコントロールユニット（以下、FCUという）65と、印刷等を行うためのエンジン70と、プロッタ75とを有している。

【0045】

コントローラ40は、CPU41と、MEM-P42と、ノースブリッジ（以下、NBという）43と、サウスブリッジ（以下、SBという）44と、ASIC45と、MEM-C46と、HDD47とを有する。また、各種インタフェースとしてNIC48、USB49、IEEE1394（50）、セントロニクス51を有している。

【0046】

オペレーションパネル 60 は、コントローラ 40 の A S I C 45 に直接接続されている。また、F C U 65、各種インタフェース及びエンジン 70、プロッタ 75 は、コントローラ 40 の A S I C 45 に P C I バスで接続されている。

【0047】

C P U 41 は、複合機 100 の全体制御を行うものである。N B 43 は、C P U 41、M E M - P 42、A S I C 45 を接続するためのブリッジである。M E M - P 42 は、複合機 100 の描画用メモリなどとして用いるシステムメモリである。S B 44 は、N B 43 と周辺デバイスなどとを接続するためのブリッジである。本実施の形態では、メモリカードスロット 441 がセットされており、メモリカード 442 を挿入できる。また、M E M - C 46 は、コピー用画像バッファ、符号バッファとして用いるローカルメモリである。A S I C 45 は、画像処理用のハードウェア要素を有する画像処理用途向けの I C である。また、H D D 47 は、画像データの蓄積、プログラムの蓄積、フォントデータの蓄積、フォームの蓄積を行うためのストレージである。

【0048】

(第 1 の実施の形態)

次に、本発明の第 1 の実施の形態として、複合機 100 における J a v a (登録商標) アプリケーション 117 の実行環境 116 について説明する。

【0049】

図 3 に、複合機 100 に搭載される J a v a (登録商標) 実行環境 116 の構成の一例を示す。図 3 に示すように、本実施の形態における J a v a (登録商標) 実行環境 116 は、複合機クラスライブラリ 201、仮想マシン 202、ネイティブプログラムインタフェース 203、及びアプリケーション管理部 204 を有する。

【0050】

複合機クラスライブラリ 201 は、J a v a (登録商標) アプリ 117 が容易に複合機 100 を操作するためのサービスを提供するために用いられるクラスライブラリであり、ネイティブプログラムインタフェース 203 の機構を用いて複合機 100 の A P I にアクセスする。この複合機クラスライブラリ 201 により、機種依存部分の区別、簡易操作機能の提供、種々のオブジェクト指向クラスの提供がなされる。

【0051】

本実施の形態における複合機クラスライブラリ 201 は、基本的な J a v a (登録商標) クラスライブラリ、イベント処理のためのイベントクラス、ネットワークに関する処理を行うネットワークランザクションクラス、複合機 100 の操作パネルへの描画等を行うための操作パネルクラス、複合機 100 を制御するための複合機制御クラス等から構成される。なお、操作パネルクラスと複合機制御クラスは本実施の形態における複合機 100 に特有のクラスである。以下、操作パネルクラスと複合機制御クラスをまとめて複合機クラスと称し、その他の基本的な J a v a (登録商標) クラスライブラリ、イベントクラス、ネットワークランザクションクラスをまとめて J a v a (登録商標) コアクラスと称する場合がある。

【0052】

仮想マシン 202 は、J a v a (登録商標) のソースコードプログラムをいったん中間コード形式にコンパイルし、その結果として得られるバイトコードを解釈・実行する機能を有し、ネイティブプログラムインタフェース 203 は、仮想マシン 202 で実行される J a v a (登録商標) コードが、C 言語などの他のプログラミング言語で書かれたアプリケーションやライブラリと相互運用するための機能を有している。

【0053】

また、アプリケーション管理部 204 は、J a v a (登録商標) アプリを管理する機能を有しており、例えば、J a v a (登録商標) アプリのリスト表示、J a v a (登録商標) アプリ実行管理、J a v a (登録商標) アプリのインストールやバージョンアップ、インストール済みの J a v a (登録商標) アプリの削除、アプリケーション登録のためのパスワード設定、J a v a (登録商標) アプリの保存等を行う機能を有している。J a v a

(登録商標) アプリ実行管理としては、アプリケーションの起動、強制終了、サスペンド、レジュームなどを行う。

【0054】

アプリケーション管理部204は、仮想マシン202から独立したコンポーネントとして実装されており、Java (登録商標) アプリ117からは制御できない。なお、図4(a)、(b)に示すように、アプリケーション管理部204は、Java (登録商標) アプリ117として実装(a)してもよいし、例えばC言語などにより実装(b)することもできる。

【0055】

なお、Java (登録商標) 実行環境116は複合機100に搭載されるアプリケーションの一種であるので、他のアプリケーションと同様の方法により複合機100にインストールすることが可能であり、例えばメモリカードから複合機100にインストールすることができる。

【0056】

(Java (登録商標) アプリの例)

次に、Java (登録商標) アプリの例について説明する。ここでは、操作パネルクラスを使用したJava (登録商標) アプリの例について説明する。図5に、このJava (登録商標) アプリで使用する操作パネルクラスの階層構成の一例を示す。また、図6に、このJava (登録商標) アプリのプログラムリストを示す。以下、このアプリケーションプログラムのことをシンプルプリンターと称する場合がある。

【0057】

図6に示すプログラムについて以下説明する。このプログラムは、複合機100の操作パネルにユーザーインタフェース画面を表示し、操作パネルから印刷ファイル名を指定し、指定したファイルを印刷するプログラムである。このプログラムの動作概要は次の通りである。

【0058】

まず、複合機100の起動直後に操作パネル上に図7に示す画面を表示する。すなわち、画面の上部に "Simple Printer" の文字列を表示し、中央部に "Get input filename..." のボタンを表示する。ユーザーが、"Get input filename..." のボタンに触れると、パネル上に、例えば図8に示すソフトキーボードを表示する。ソフトキーボードには、"Input file name" のタイトルが表示され、ユーザーは、ソフトキーボードから印刷したいファイル名を入力する。例えば、"/usr/hdd/photo.tiff" と入力した場合、"/usr/hdd" のディレクトリに格納されている "photo.tiff" ファイルを印刷する。

【0059】

このプログラムについて、プログラム中に記述した注釈の番号に沿って説明する。

【0060】

(1) は、複合機クラスライブラリを使用するための命令である。(2) は、複雑な初期設定や終了処理に関する複合機のアプリケーションの雛型 (抽象) である GApp を継承することを示している。これにより、複雑な初期設定や終了処理をユーザーが記述せずに済み、メッセージ受信などの処理もユーザーに対して隠蔽する。

【0061】

(3) によりプリンターを操作するオブジェクトを生成し、(4) により "Simple Printer" のメッセージを表示するオブジェクトを生成し、(5) により "Get input filename..." のボタンオブジェクトを生成する。(6) は、ボタンに触れるとソフトキーボードが表示されるようにオブジェクトを追加することを示している。(7) は、ソフトキーボードを表示し、ファイル名入力を促すことを示している。(8) により、ソフトキーボードから入力されたファイルを印刷する。

【0062】

なお、プログラムのボタン操作 ((5) "Get input filename...") では、ボタン操作の結果は、プログラムがメッセージとして受信し、そのメッセージを受信した場合に実際

のボタン操作に対応した処理を行う。その処理が（６）のaddChangeListener()で設定されている。

【0063】

（ダウンロードから起動までの手順の例）

J a v a（登録商標）アプリのインストール及び起動はアプリケーション管理部 2 0 4 が管理しており、アプリケーション管理部 2 0 4 の機能を用いることにより、以下のような手順で J a v a（登録商標）アプリのダウンロードから起動までの処理を行うことが可能である。この手順を図 9 のフローチャートを参照しながら、上記のシンプルプリンターを起動する場合を例にとり説明する。

【0064】

アプリケーション管理部 2 0 4 は、他のアプリケーションと同様にして複合機 1 0 0 の起動時に起動される。アプリケーション管理部 2 0 4 に対応するアプリケーション選択キーを押すことにより、複合機 1 0 0 の操作パネルにアプリケーション管理部 2 0 4 を操作するためのユーザーインタフェース画面を表示する（ステップ S 1）。当該ユーザーインタフェース画面において J a v a（登録商標）アプリとしてシンプルプリンターがインストールされていることがわかれば、シンプルプリンターを選択して起動する（ステップ S 2）。

【0065】

インストールされていなければ、ユーザーインタフェース画面からアプリケーションのロードを指定することによりアプリケーションロード画面が表示される（ステップ S 3）。

【0066】

ユーザーは、このアプリケーションロード画面から、Web サイトからシンプルプリンターをダウンロードするか、複合機 1 0 0 に挿入されるメモリカードからダウンロードするかを選択する（ステップ S 4）。

【0067】

メモリカードからダウンロードする場合は、メモリカードを複合機 1 0 0 に差し込むことによりアプリケーション管理部 2 0 4 がメモリカード内のファイルを調査し、格納されているファイルを複合機 1 0 0 の操作パネル上に表示する（ステップ S 5）。ユーザーがシンプルプリンターを選択すると（ステップ S 6）、アプリケーション管理部 2 0 4 は当該アプリが複合機 1 0 0 で実行可能かどうかの確認を行う（ステップ S 7）。そして、確認の結果、実行可能であればダウンロードを行い、ハードディスクへのインストールを行うとともに、アプリケーション管理部 2 0 4 は、アプリケーション名（シンプルプリンター）を取得し、アプリケーションリストに追加する（ステップ S 9）。なお、アプリケーションリストは、インストール済みの J a v a（登録商標）アプリを把握するために、アプリケーション管理部 2 0 4 が管理するファイルである。

【0068】

Web サーバーからシンプルプリンターをダウンロードする場合は、ステップ 4 にて Web を選択すると、URL を指定するための画面が表示され、そこでユーザが、シンプルプリンターを保持する Web サイトの URL を入力する（ステップ S 10）。アプリケーション管理部 2 0 4 が Web サイトにアクセスし（ステップ S 11）、シンプルプリンターが複合機 1 0 0 で実行可能かどうかの確認を行い（ステップ S 12）、シンプルプリンターをダウンロードする（ステップ S 13）。そして、アプリケーション管理部 2 0 4 は、ハードディスクへのインストールを行うとともに、アプリケーション名（シンプルプリンター）を取得し、アプリケーションリストに追加する（ステップ S 14）。

【0069】

なお、アプリケーション管理部 2 0 4 が自動的にメモリカード又は Web サーバーを検索し、ハードディスクへのインストールを行うようにしてもよい。なお、上記の例では複合機 1 0 0 に J a v a（登録商標）アプリをインストールする例について説明しているが、J a v a（登録商標）アプリは必ずしも複合機 1 0 0 のハードディスクにインストール

しなくても実行することができる。

【0070】

上記の J a v a (登録商標) アプリが複合機 100 で実行可能か否かに関する確認に際しては、アプリケーション管理部 204 は、プログラムサイズ、バージョン、プログラム最終更新情報、使用するメモリワークサイズ、使用するストレージサイズ、予め使用するネットワークアドレス、利用可能期間(利用回数)、アプリケーションプログラム名等についての確認を行う。これらの情報は、J a v a (登録商標) アプリに付随している情報である。なお、第2の実施の形態において後述するように、これらの情報は、J a v a (登録商標) アプリに対応する J N L P ファイルから取得することが可能である。

【0071】

以上の処理が完了することにより、ダウンロードしたシンプルプリンターを利用できる状態になり、ユーザーは、ユーザーインターフェース画面(ステップ S1) からシンプルプリンターを起動する(ステップ S15)。これにより、仮想マシン 202 が起動し、シンプルプリンターが実行される。

【0072】

起動されたシンプルプリンターは J a v a (登録商標) コアクラス及び複合機クラスを使用して複合機 100 を操作し、前述した通りのシンプルプリンター用のユーザーインターフェース画面を表示する。ユーザーは、このユーザーインターフェース画面から印刷したいファイル名を入力し、印刷を行う。ここでは、"/usr/hdd/photo.tiff" と入力を行い、複合機内のハードディスク内の "/usr/hdd" のディレクトリに格納されている "photo.tiff" ファイルを印刷する。

【0073】

印刷が完了すると、アプリケーション管理部 204 からシンプルプリンターの実行を終了する。

(第2の実施の形態)

複合機 100 では、J N L P (J a v a (登録商標) Networking Language Protocol) を用いて、インストールされている J a v a (登録商標) アプリの自動更新を行うことが可能である。以下、J N L P を用いた J a v a (登録商標) アプリの自動更新について説明する。なお、以下の処理はアプリケーション管理部 204 の制御により行われるものである。

【0074】

以下で説明する処理では J N L P ファイル(アプリケーション情報ファイルともいう) が用いられるので、まず、J N L P ファイルについて説明する。J N L P ファイルは J a v a (登録商標) アプリと 1 対 1 に対応しており、J a v a (登録商標) アプリの定義付けに係る情報を含んでいる。すなわち、J a v a (登録商標) アプリ毎に、それに対応する J N L P ファイルが付随する。

【0075】

図 10 は、J N L P ファイルの構文の例を表す。J N L P ファイルは XML (e X t e n s i b l e M a r k u p L a n g u a g e) ファイルであり、J N L P ファイル書式は J N L P 規格に準拠している。ただし、本実施の形態では独自の拡張を施している部分があるため、以下その部分について説明する。

【0076】

記述 1 は、i n f o r m a t i o n エレメントであり、アプリケーション名を示す t i t l e エレメント(記述 1 A) や、ベンダ名を示す v e n d e r エレメント(記述 1 B) や、ベンダの電話番号を示す t e l e p h o n e エレメント(記述 1 C) や、ベンダのファクス電話番号を示す f a x エレメント(記述 1 D) や、アプリケーションのプロダクト ID を示す p r o d u c t I D エレメント(記述 1 E) を含んでいる。

【0077】

記述 2 は、s e c u r i t y エレメントである。

【0078】

記述3は、resourceエレメントであり、Java（登録商標）アプリのバージョンを指定するjdkエレメント（記述3A）や、JARファイル（Java（登録商標）アプリの実行ファイル）とそのバージョンを指定するjarエレメント（記述3B）や、SUB-JNLPファイルを指定するsub-jnlpエレメント（記述3C）を含んでいる。

【0079】

記述4は、updateエレメントであり、更新処理の実行方法を設定するエレメントである。autoであればアプリケーションの更新処理は自動更新で実行され、manualであればアプリケーションの更新処理は手動更新で実行され、mailであればアプリケーションの更新処理が実行可能である場合にその旨を通知する更新通知メールが配信される。

【0080】

記述5は、installエレメントであり、インストール処理の実行方法を設定するエレメントである。autoであればアプリケーションのインストール先は自動選択にて選択され、manualであればアプリケーションのインストール先は手動選択にて選択される。記述6は、アプリケーションのタイプが「GUI有」であるか「GUI無」であるかを示す。

【0081】

また、Java（登録商標）アプリに対応するJNLPファイルの内容は、一般的に図11に示すようなツリー構造となっている。上位のJNLPファイルから参照されるJNLPファイルは、当該上位のJava（登録商標）プログラムの実行の中で使用されるモジュールに対応している。

【0082】

図10のupdateエレメントで触れたように、インストール済みのJava（登録商標）アプリは更新処理の対象となる。以下、インストール済みのJava（登録商標）アプリを自動更新する自動更新処理について説明する。

【0083】

図12は、メモ리카ードの一例であるSDcard又はHDD（ハードディスク）内のJava（登録商標）アプリ（更新対象アプリ）を、SDcardからのJava（登録商標）アプリ（更新用アプリ）に自動更新する自動更新処理に係るフローチャートである。

【0084】

アプリケーション管理部204は、複合機100に接続された更新用SDcardスロットに更新用SDcardがセットされた場合、更新用SDcard内の更新用アプリのJNLPファイルを取得（ステップS101）する。アプリケーション管理部204は、そのJNLPファイルのupdateエレメントが「auto」で、そのJNLPファイルに係る更新用アプリと同一の更新対象アプリが複合機100に存在する場合（ステップS102・S103）には、その更新用アプリのバージョンとその更新対象アプリのバージョンとを比較（ステップS104）する。アプリケーション管理部204は、更新用アプリのバージョンが更新対象アプリのバージョンより新しい場合（ステップS105）には、更新用アプリをもって更新対象アプリを更新（ステップS106）する。すなわち、JarファイルとJNLPファイルを更新する。

【0085】

図13は、SDcard又はHDD内のJava（登録商標）アプリ（更新対象アプリ）を、WebからのJava（登録商標）アプリ（更新用アプリ）に自動更新する自動更新処理に係るフローチャートである。

【0086】

アプリケーション管理部204は、更新対象アプリのJNLPファイルを取得（ステップS201）する。アプリケーション管理部204は、そのJNLPファイルのupdateエレメントが「auto」で、そのJNLPファイルに係る更新対象アプリと同一の

更新用アプリが、複合機 1 0 0 とネットワーク接続された W e b サーバ内に存在する場合（ステップ S 2 0 2 ・ S 2 0 3）には、その更新対象アプリのバージョンとその更新用アプリのバージョンとを比較（ステップ S 2 0 4）する。アプリケーション管理部 2 0 4 は、更新対象アプリのバージョンより更新用アプリのバージョンが新しい場合（ステップ S 2 0 5）には、更新対象アプリを更新用アプリをもって更新（ステップ S 2 0 6）する。

【 0 0 8 7 】

なお、複合機 1 0 0 が更新用アプリにアクセスする先は J N L P ファイルを参照することにより取得できる。また、J N L P ファイルがツリー構造である場合には、各階層の J N L P ファイルに対して上記の処理がなされる。

【 0 0 8 8 】

また、上記の更新のトリガーは、例えば、複合機 1 0 0 電源 O N からの経過時間、所定の時刻、電源 O N / 電源 O F F、アプリケーションの実行直前 / 実行中 / 実行後、アイドルの所定時間経過、等とすることができる。

【 0 0 8 9 】

第 1 の実施の形態においても J N L P ファイルを用いる構成とすることができる。その場合、J a v a（登録商標）アプリのダウンロードの前に、それに対応する J N L P ファイルを取得し、その J N L P の記述に基づき当該 J a v a（登録商標）アプリが実行可能かどうかを確認することができる。その場合、J N L P ファイルには、J a v a（登録商標）アプリが複合機 1 0 0 で実行可能かどうかを判断するための情報が記述される。例えば、第 1 の実施の形態で説明したプログラムサイズ等の他、認証のための情報、複合機 1 0 0 側で J a v a（登録商標）アプリ実行のために必要な機器構成（カラー対応等）、使用リソース情報等が記述される。この J N L P ファイルに記述された情報と、複合機 1 0 0 の使用リソース情報等とを比較した結果、J a v a（登録商標）アプリを実行できないと判断したら、例えば、警告を表示する。また、認証のために、J a r ファイルに電子署名を付し、アプリケーション管理部 2 0 4 が署名の検証を行うようにしてもよい。

【 0 0 9 0 】

（第 3 の実施の形態）

次に、P C などにより開発した J a v a（登録商標）アプリを当該 P C で実行することにより、複合機をリモート操作する方法について説明する。

【 0 0 9 1 】

図 1 4 に、本実施の形態における J a v a（登録商標）アプリ実行時のシステム構成を示す。同図に示すように、本システムは、複合機の機能を有していない P C 3 0 0 と、複合機 1 0 0 とがネットワーク 4 0 0 により接続された構成を有している。P C 3 0 0 は、例えば、複合機 1 0 0 用の J a v a（登録商標）アプリの開発のための P C である。

【 0 0 9 2 】

P C 3 0 0 は、J a v a（登録商標）アプリケーションであるシンプルプリンター 3 0 1、J a v a（登録商標）コアクラス 3 0 4、複合機 1 0 0 における複合機クラスを操作することを可能とするクライアント複合機クラス 3 0 3、複合機 1 0 0 のアプリケーション管理部 2 0 4 と通信することによりアプリケーションの管理を行うリモートアプリケーション管理部 3 0 2、仮想マシン 3 0 5、及び O S 3 0 6 を有している。

【 0 0 9 3 】

また、複合機 1 0 0 は、J a v a（登録商標）実行環境 1 1 6 の構成に加えて、サーバ複合機アプリケーション 2 1 0 を有している。サーバ複合機アプリケーション 2 1 0 は、P C 3 0 0 からのリモート操作サービスを提供するために複合機 1 0 0 で実行されるアプリケーションである。なお、図 1 4 ではネイティブプログラムインタフェースの図示を省略している。

【 0 0 9 4 】

P C 3 0 0 におけるクライアント複合機クラス 3 0 3 は、サーバ複合機アプリケーション 2 1 0 と通信を行い、複合機 1 0 0 のオブジェクトを操作することにより複合機 1 0 0 を P C 3 0 0 のアプリケーションからリモート操作することを可能とする機能を有してい

る。これにより、アプリケーションの分散処理を実現する。遠隔にあるPC300から複合機100上のオブジェクトを操作する技術は、例えばJava（登録商標）ベースの分散処理技術を用いることにより実現できる。

【0095】

また、リモートアプリケーション管理部302が複合機100に搭載されたアプリケーション管理部204と通信することにより、複合機100用のJava（登録商標）アプリケーション管理機能をPC300まで拡張することが可能である。なお、リモートアプリケーション管理部302は、複合機100のアプリケーション管理部204と同様の機能を有している。すなわち、Java（登録商標）アプリ実行管理機能（起動/強制終了/サスペンド/レジュームなど）、Java（登録商標）アプリのインストールやバージョンアップ（ダウンロード時のチェック）、インストール済みのJava（登録商標）アプリの削除、アプリケーション登録のパスワード設定、Java（登録商標）アプリケーションの保存、インストールされているJava（登録商標）アプリのリスト表示などの機能を有している。上記の処理を、複合機100のアプリケーション管理部204と必要に応じて通信を行いながら実行する。

【0096】

また、PC300のクライアント複合機クラス303は、複合機100の一部の機能をエミュレートする機能も有している。エミュレーションの機能としては、例えば操作パネルのエミュレーションがある。

【0097】

すなわち、PC300で実行するアプリケーションにより複合機100を操作する場合、複合機100の操作パネルを使用するのか、それとも、PC300で操作パネルをエミュレーションして表示するのかを選択できる。操作パネルの機能をエミュレートする場合、複合機100の操作パネルにおける画面表示は、PC300の画面上への表示とし、複合機100の操作パネルにおける操作ボタンの操作には、PC300のキーボード及びマウスを使用する。なお、複合機100の操作パネルを使用するのか複合機100の操作パネルを使用するのかの選択は、リモートアプリケーション管理部302から複合機100のアプリケーション管理部204に要求を出すことにより行われる。例えば、リモートアプリケーション管理部302から複合機100のアプリケーション管理部204にPC300で操作パネルをエミュレートする旨の通知をした場合には、アプリケーション管理部204は、複合機100の操作により生じる画面表示情報を複合機100に表示するのではなくPC300に送信するように制御する。

【0098】

また、Java（登録商標）アプリをPC300から実行する場合に、リモート側のPC300のリソースと複合機100のリソースで共通するリソースに関してはいずれかのリソースを選択することができる。例えば、PC300と複合機100で共通するリソースであるハードディスクへのアクセスを指示する場合、PC300のハードディスクと複合機100のハードディスクのどちらを使用するかをPC300から選択できる。そのような共通のリソースとして、ネットワーク機能に関するリソースやI/O（USB、link、RS232C）などがある。

【0099】

次に、図14に示す環境において、Java（登録商標）アプリをPC300で実行して複合機100を操作する手順について図15のフローチャートを参照して説明する。以下、Java（登録商標）アプリとして上述したシンプルプリンターを用いる例について説明する。

【0100】

まず、PC300においてリモートアプリケーション管理部302を起動する。なお、複合機100では既にアプリケーション管理部204を含むJava（登録商標）実行環境116とサーバ複合機アプリケーション210が実行されているものとする。

【0101】

リモートアプリケーション管理部302は、アプリケーション選択用の画面をPC300上に表示する(ステップS21)。このときJava(登録商標)アプリが複合機100のハードディスク又はPC300のハードディスクにインストールされているか否かをチェックする。このチェックは、例えば、リモートアプリケーション管理部302がPC300での確認を行い、複合機100のアプリケーション管理部204が複合機100での確認を行ってリモートアプリケーション管理部302に結果を通知することにより実施することができる。Java(登録商標)アプリが存在すると、前記画面にJava(登録商標)アプリ名を表示する。ユーザがシンプルプリンターを選択することにより、シンプルプリンターが起動される(ステップS22)。

【0102】

Java(登録商標)アプリがインストールされているか否かの確認の結果、どちらのハードディスクにもJava(登録商標)アプリがインストールされていない場合には、上記の画面からアプリケーションのロードを指定することによりアプリケーションロード画面が表示される(ステップS23)。

【0103】

ユーザーは、このアプリケーションロード画面から、Webサイトからシンプルプリンターをダウンロードするか、複合機100に挿入されるメモリカードからダウンロードするかを選択する(ステップS24)。

【0104】

メモリカードからJava(登録商標)アプリをインストールする場合、メモリカードを複合機100に差し込むことによりアプリケーション管理部204がメモリカードを認識し、メモリカード内のファイル情報を調査し、ファイル情報をリモートアプリケーション管理部302に通知する(ステップS25)。

【0105】

リモートアプリケーション管理部302は、通知されたファイル情報をPC300の画面に表示し(ステップS26)、ユーザーにシンプルプリンターを選択させる(ステップS27)。そして、リモートアプリケーション管理部302はシンプルプリンターをダウンロードする旨をアプリケーション管理部204に通知し、アプリケーション管理部204はアプリケーション(シンプルプリンター)に関する情報をリモートアプリケーション管理部302に送信する。リモートアプリケーション管理部302はシンプルプリンターが本システムの環境で実行可能であるか否かを確認し(ステップS28)、確認がとれたらアプリケーション管理部204にシンプルプリンターのダウンロードの指示をすることにより、シンプルプリンターのダウンロードがなされ(ステップS29)、予め指定されたインストール先(PC300のハードディスク又は複合機100のハードディスク)にインストールされる(ステップS30)。また、アプリケーションリストへシンプルプリンターが追加される。

【0106】

なお、上記ステップS28における確認項目は、例えば、プログラムサイズ、バージョン、プログラム最終更新情報、使用するメモリワークサイズ、使用するストレージサイズ、あらかじめ使用するネットワークアドレスの確認、利用可能期間(利用回数)、アプリケーションプログラム名等である。

【0107】

Webサイトからシンプルプリンターをインストールする場合は、WebサーバーのURLを指定させる画面をリモートアプリケーション管理部302が表示し、その画面からユーザーがURLを入力する(ステップS31)。そして、リモートアプリケーション管理部302がURLを複合機100のアプリケーション管理部204に通知し(ステップS32)、アプリケーション管理部204がURLにより指定されたWebサイトにアクセスし(ステップS33)、当該Webサイトが保持するアプリケーション(シンプルプリンター)がPC300から複合機100を操作可能かどうかを確認する(ステップS34)。なお、この確認は、所定の情報(例えばJNLPファイル)をアプリケーション管

理部 2 0 4 が取得し、それをリモートアプリケーション管理部 3 0 2 に送信することにより、リモートアプリケーション管理部 3 0 2 が行うようにしてもよい。

【0 1 0 8】

確認がとればシンプルプリンターをダウンロードし（ステップ S 3 5）、指定されたハードディスクにインストールするとともにアプリケーションリストにシンプルプリンターを追加する（ステップ S 3 6）。

【0 1 0 9】

シンプルプリンターがインストールされると、ステップ S 2 1 における画面からシンプルプリンターを選択することができる。選択がなされると、リモートアプリケーション管理部 3 0 2 によりシンプルプリンターが P C 3 0 0 側の仮想マシン 3 0 5 上にロードされ、実行される（ステップ S 3 7）。

【0 1 1 0】

シンプルプリンターは、J a v a（登録商標）コアクラス 3 0 3 及びクライアント複合機クラス 3 0 3 を使用して複合機 1 0 0 を操作する。複合機 1 0 0 の操作にあたっては、例えば、シンプルプリンター 3 0 1 から複合機 1 0 0 のサーバ複合機アプリケーション 2 1 0 にメッセージを送信し、そのメッセージに基づきサーバ複合機アプリケーション 2 1 0 が複合機 1 0 0 の J a v a（登録商標）実行環境におけるオブジェクトを操作することにより複合機 1 0 0 を操作する。

【0 1 1 1】

ここで、シンプルプリンター 3 0 1 は、前述したように始めにグラフィカルインターフェース画面を表示するように動作するが、その画面を P C 3 0 0 に表示するか、複合機 1 0 0 の操作パネルに表示するかを選択することができる。この選択は、例えば、ユーザーがシンプルプリンターを選択した後に、これらのうちいずれかを選択させる画面をリモートアプリケーション管理部 3 0 2 が表示することにより実現できる。ここでは、P C 3 0 0 を選択し、P C 3 0 0 に操作パネルの画面をエミュレートした画面を表示する。

【0 1 1 2】

次に、ユーザーは P C 3 0 0 の画面から印刷するファイル名を選択することになるが、ここでは、ファイルの指定先を、P C 3 0 0 のハードディスクにするか、複合機 1 0 0 のハードディスクにするかを選択できる。複合機 1 0 0 のハードディスクを選択できることにより、あたかも複合機 1 0 0 の操作パネルで操作しているかのような操作を P C 3 0 0 上で行うことが可能となる。この指定は、例えば、リモートアプリケーション管理部 3 0 2 によりシンプルプリンターを選択したときに、どちらのハードディスクを選択するかを入力させる画面を表示することにより実現できる。

【0 1 1 3】

そして、複合機 1 0 0 のハードディスクを指定した場合は、P C 3 0 0 の画面から、印刷するファイルを例えば"/hdd/ts/photo.tiff"と指定すると、複合機 1 0 0 のハードディスクにあるファイルが選択され、印刷が実行される。

【0 1 1 4】

複合機 1 0 0 の操作を完了したい場合は、リモートアプリケーション管理部 3 0 2 は、複合機 1 0 0 のアプリケーション管理部 2 0 4 に終了又は休止の要求を行い、複合機 1 0 0 との通信を遮断する。

【0 1 1 5】

なお、第 3 の実施の形態においても、第 2 の実施の形態のように、J N L P ファイルを用いて J a v a（登録商標）アプリの自動更新を行うことが可能である。

【0 1 1 6】

（第 4 の実施の形態）

第 3 の実施の形態における遠隔操作に関する他の例を第 4 の実施の形態として次に説明する。第 4 の実施の形態では、複合機 1 0 0 において、図 1 に示した J A V A（登録商標）実行環境 1 1 6、J A V A（登録商標）アプリ 1 1 7 の部分の構成を、図 1 6 に示す構成としている。

【0117】

図16に示すように、J SDKアプリ147とJ SDKプラットフォーム148を備えている。J SDKプラットフォーム148は、他のソフトウェアとの仲介やアプリケーション管理を行うものである。

【0118】

SDKとは、専用のソフトウェア開発キットであり、SDKを使用して開発したアプリケーションをSDKアプリと呼ぶ。専用のSDKとしては、例えばJava（登録商標）言語でアプリケーションを開発するための「J SDK」が提供される。J SDKを使用して開発したアプリケーションを「J SDKアプリ」と呼ぶ。

【0119】

図17は、図16のJ SDKアプリ147とJ SDKプラットフォーム148のクラス図である。J SDKアプリ147とJ SDKプラットフォーム148は、Java（登録商標）コンパイラによりソースコードからバイトコードに一括翻訳されており、Java（登録商標）仮想マシンにより逐次実行される。J SDKアプリ147とJ SDKプラットフォーム148は、Java（登録商標）2 Micro EditionのPersonal Basis Profileをベースとする実装となっている。

【0120】

J SDKアプリ147としては、ユーザアプリ501と、J SDK GUI Manager511と、Task Bar Manager512等が存在する。

【0121】

ユーザアプリ501は、複合機100のユーザ（例えばベンダ）がJ SDKを使用して開発したJ SDKアプリである。J SDK GUI Manager511は、他のJ SDKアプリ（ユーザアプリ501等）を操作対象とする操作画面の表示等を行うJ SDKアプリである。Task Bar Manager512は、他のJ SDKアプリ（ユーザアプリ501等）を操作対象とするタスクバーの表示等を行うJ SDKアプリである。

【0122】

ユーザアプリ501はここでは、スタンドアロンアプリケーションやアプレットと並ぶJava（登録商標）アプリケーションであるXletである。J SDK GUI Manager511とTask Bar Manager512はここでは、独自の拡張を施したXlet（XletEx）である。

【0123】

J SDKプラットフォーム148には、J SDK Main521と、J SDK Environment522と、Locale Manager523と、Xlet Manager531と、Multi Xlet Manager532と、J SDK Manager533と、Send Manager541と、Event Manager542と、System Event Manager543と、Panel Manager544と、Install Manager545と、Server/Client Manager546等のクラスが存在する。

【0124】

J SDK Main521は、J SDK Environment522の起動を行うクラスである。J SDK Environment522は、J SDKシステムの起動環境設定を行うクラスである。Locale Manager523は、国際化対応（言語指定）を行うクラスである。

【0125】

Xlet Manager531は、1対1でXletのライフサイクルを管理するクラスである。ここでは、5個のXletのライフサイクルが1対1で5個のXlet Manager531によって管理される。Multi Xlet Manager532は、全てのXlet Manager531のライフサイクルを管理するクラスである。ここでは、5個のXlet Manager531のライフサイクルが全て1個のMulti Xlet Manager532によって管理される。J SDK Manager

533は、JSDKシステム全体のライフサイクルを管理するクラスである。例えば、MultiXletManager532, SendManager541, EventManager542, SystemEventManager543, PanelManager544, InstallManager545, Server/ClientManager546のライフサイクルが、JSDKManager533によって管理される。なお、Xletのライフサイクルとは、初期化状態(Loaded)、停止状態(Paused)、活性化状態(Active)、終了状態(Destroyed)等の状態の遷移のことである。

【0126】

SystemEventManager543は、図1のプラットフォーム120からのシステムイベント(電力モード等)の管理を行うクラスである。PanelManager544は、1個のXletがオペレーションパネルの画面を占有する際の調停等を行うクラスである。InstallManager545は、SDcardやWebからのインストールやアンインストールの管理を行うクラスである。

【0127】

図17のJSDKシステムでは、APIとして、JSDKAPI551とJSDKAPI552が利用される。なお、XletとXletExの差異として、オブジェクトにアクセスするのにJSDKAPI551の利用とJSDKプラットフォーム148へのアクセスが可能である点が挙げられる。図16の複合機100にはさらに、図17のJSDKシステムに係る要素として、C言語とJava(登録商標)言語のインタフェースとなるJSDKSession553とNativeJSDKSession554や、JSDKアプリ147とJSDKプラットフォーム148を実行するためJava(登録商標)仮想マシンであるCVM555等が存在する。

【0128】

このような構成を有する複合機100を、図14に示した場合と同様にして、図18に示すように、PC701と接続する。PCは複合機100のエミュレータとして機能する。

【0129】

図18のPC701は、例えばPC本体711と、キーボード712と、マウス713と、ディスプレイ714等により構成される。PC本体711には、CPU、ROM、RAM、NVRAM、HDD、MODEM、NIC等が存在する。キーボード712とマウス713は、複合機100のオペレーションパネルに代わって、オペレータが入力を行うためのハードウェア(操作部)となる。ディスプレイ714は、複合機100のオペレーションパネルに代わって、オペレータが出力を得るためのハードウェア(表示部)となる。

【0130】

図19は、図18のPC701内のJSDKアプリ147とJSDKプラットフォーム148のクラス図である。図16の複合機100と同様のJSDKアプリ147とJSDKプラットフォーム148が、図19のPC701内にも存在する。図16の複合機100内のJSDKアプリ147とJSDKプラットフォーム148のクラス図については、図17を参照されたい。以下、図17と図19の差異について説明する。

【0131】

第1に、図17のPanelManager544は、図19のPanelManagerEmulator744に置換されている。第2に、図17のJSDKSession553は、図19のEmulatorJSDKSession753に置換されている。第3に、図17のNativeJSDKSession554は、図19のEventEmulator754に置換されている。第4には、図17のCVM555は、図19のJVM755に置換されている。

【0132】

PanelManagerEmulator744は、PanelManage

r 5 4 4 をエミュレートして、オペレーションパネルの操作をキーボード 7 1 2 やマウス 7 1 3 の操作に変換するクラスである。Emulator J SDK Session 7 5 3 は、通信経路の確立処理を実行するクラスである。Event Emulator 7 5 4 は、複合機 1 0 0 の動作をエミュレートするクラスである。JVM 7 5 5 は、JSD K アプリ 1 4 7 と J SDK プラットフォーム 1 4 8 を実行するため J a v a (登録商標) 仮想マシンである。

【0133】

図 2 0 は、複合機 1 0 0 内で実行される遠隔操作の準備処理の実行手順について説明するための図である。

【0134】

最初に、複合機 1 0 0 内の Server/Client Manager 5 4 6 と PC 7 0 1 内の Server/Client Manager 5 4 6 との間で、セッションが確立 (1) される。なお、セッションの確立は、SIP (Session Initiation Protocol) に類似したプロトコルによるメッセージ送受信により行われる。

【0135】

次に、複合機 1 0 0 内の Server/Client Manager 5 4 6 は、遠隔操作に移行できるか否かを複合機 1 0 0 内の J SDK Manager 5 3 3 に問い合わせ、遠隔操作に移行できるか否かを判断 (2) する。

【0136】

次に、複合機 1 0 0 内の Server/Client Manager 5 4 6 は、遠隔操作に移行できる場合には、遠隔操作に必要なメッセージに係る要求を、複合機 1 0 0 内の Send Manager 5 4 1 を介して PC 7 0 1 に送信 (3) して、遠隔操作に必要なメッセージに係る応答を、複合機 1 0 0 内の Event Manager 5 4 2 を介して PC 7 0 1 から受信 (4) する。

【0137】

次に、複合機 1 0 0 内の Server/Client Manager 5 4 6 は、遠隔操作の情報を PC 7 0 1 に転送する旨の要求を、Panel Manager 5 4 4 に送信 (5) する。

【0138】

最後に、複合機 1 0 0 内の Server/Client Manager 5 4 6 と PC 7 0 1 内の Server/Client Manager 5 4 6 との間で、遠隔操作準備処理が完了した旨の通知が授受 (6) される。これによって、PC 7 0 1 からの複合機 1 0 0 の遠隔操作が可能になる。なお、第 3 の実施の形態で説明したアプリケーションの分散処理に係る機能は、例えば、複合機 1 0 0 内の Server/Client Manager 5 4 6 と PC 7 0 1 内の Server/Client Manager 5 4 6 に持たせることができる。

【0139】

第 1 ～ 第 4 の実施の形態で説明した各プログラムは、メモ리카ードなどの記録媒体に格納することができ、そのメモ리카ードから複合機 1 0 0 にインストールすることが可能である。また、PC 3 0 0 における仮想マシンなどの J a v a (登録商標) 実行環境を構成するプログラムも CD-ROM などの記録媒体に格納することができ、その CD-ROM から PC 3 0 0 にインストールすることが可能である。

【0140】

なお、本発明の実施例は上記のものに限られず、特許請求の範囲に記載の範囲において様々な応用が考えられ得ることは言うまでもない。

【図面の簡単な説明】

【0141】

【図 1】 本発明の実施の形態における複合機の構成を示すブロック図である。

【図 2】 複合機 1 0 0 のハードウェア構成図である。

【図 3】 複合機 1 0 0 に搭載される J a v a (登録商標) 実行環境 1 1 6 の構成の一

例を示す図である。

【図 4】アプリケーション管理部 204 の実装方法を説明するための図である。

【図 5】Java（登録商標）アプリで使用する操作パネルクラスの階層構成の一例を示す図である。

【図 6】シンプルプリンターのプログラムリストである。

【図 7】シンプルプリンターにより表示される画面である。

【図 8】シンプルプリンターにより表示されるソフトキーボードの例を示す図である。

【図 9】本発明の第 1 の実施の形態においてシンプルプリンターを起動するまでの手順を示すフローチャートである。

【図 10】本発明の第 2 の実施の形態における JNL P ファイルの構文の例を示す図である。

【図 11】JNL P ファイルにおけるツリー構造を示す図である。

【図 12】Java（登録商標）アプリをSD CardからのJ SDKアプリに自動更新する自動更新処理に係るフローチャートである。

【図 13】Java（登録商標）アプリをWebからのJ SDKアプリに自動更新する自動更新処理に係るフローチャートである。

【図 14】本発明の第 3 の実施の形態におけるシステム構成図である。

【図 15】本発明の第 3 の実施の形態においてシンプルプリンターを起動するまでの手順を示すフローチャートである。

【図 16】本発明の第 4 の実施の形態における複合機の構成を示すブロック図である。

【図 17】図 16 の J SDK アプリと J SDK プラットフォームのクラス図である。

【図 18】本発明の第 4 の実施の形態における接続構成図である。

【図 19】PC 内の J SDK アプリと J SDK プラットフォームのクラス図である。

【図 20】複合機内で実行される遠隔操作の準備処理の実行手順について説明するための図である。

【符号の説明】

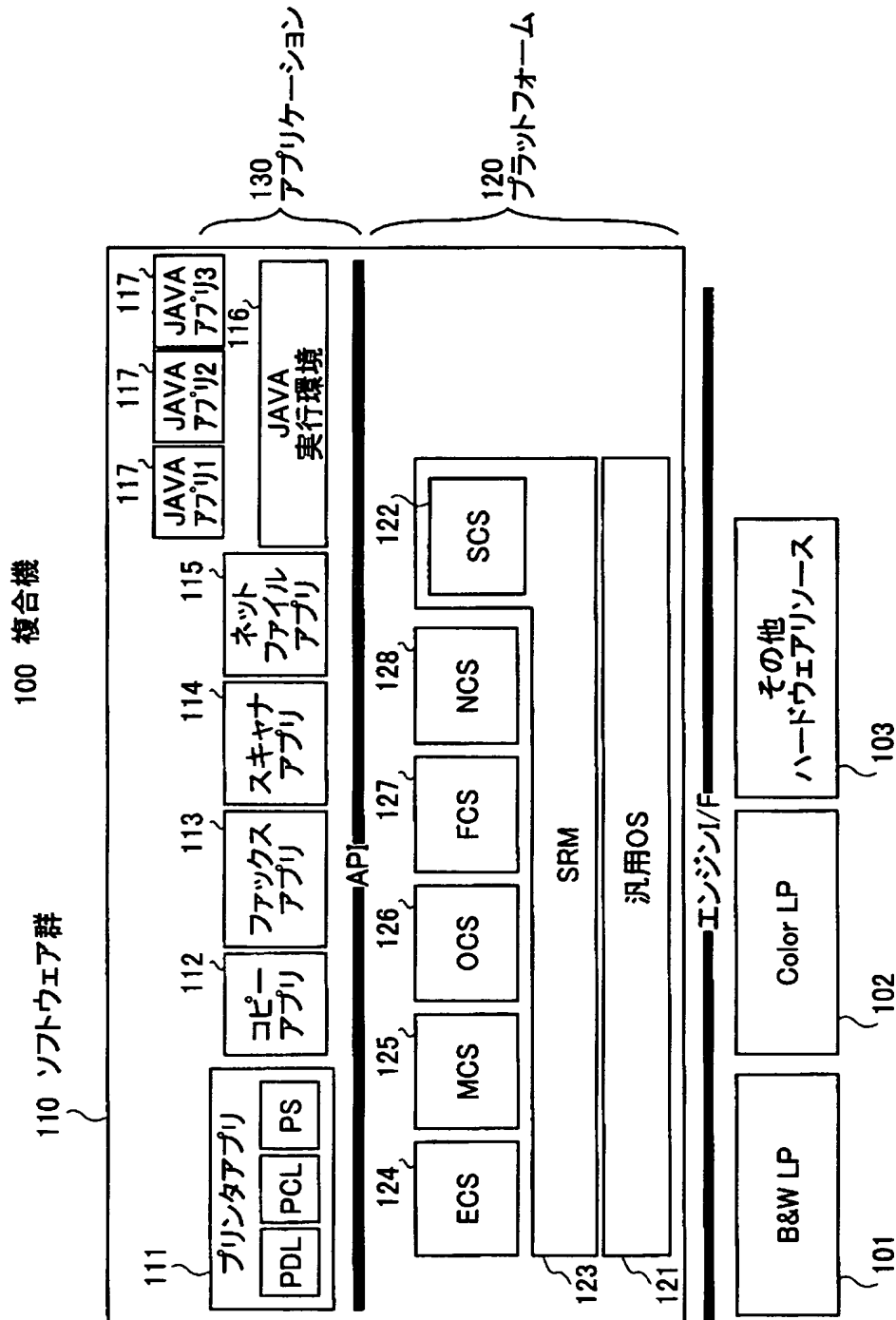
【0142】

- 100 複合機
- 101 白黒ラインプリンタ
- 102 カラーラインプリンタ
- 110 ソフトウェア群
- 111 プリンタアプリ
- 112 コピーアプリ
- 113 ファックスアプリ
- 114 スキャナアプリ
- 115 ネットファイルアプリ
- 116 Java（登録商標）実行環境
- 117 Java（登録商標）アプリ
- 120 プラットホーム
- 121 汎用OS
- 122 SCS
- 123 SRM
- 124 ECS
- 125 MCS
- 126 OCS
- 127 FCS
- 128 NCS
- 130 アプリケーション

147 JSDKアプリ
148 JSDKプラットフォーム
40 コントローラ
45 ASIC
60 オペレーションパネル
65 FCU
70 エンジン
75 プロッタ
201 複合機クラスライブラリ
202、305 仮想マシン
204 アプリケーション管理部
210 サーバ複合機アプリケーション
300 PC
301 シンプルプリンター
302 リモートアプリケーション管理部
303 クライアント複合機クラス
304 Java (登録商標) コアクラス
501 ユーザアプリ
511 JSDK GUI Manager
512 Task Bar Manager
521 JSDK Main
522 JSDK Environment
523 Local Manager
531 Xlet Manager
532 Multi Xlet Manager
533 JSDK Manager
541 Send Manager
542 Event Manager
543 System Event Manager
544 Panel Manager
545 Install Manager
546 Server/Client Manager
547 Authentication Manager
551 JSDK API
552 JSDK API
553 JSDK Session
554 Native JSDK Session
555 CVM
701 PC
711 PC本体
712 キーボード
713 マウス
714 ディスプレイ
744 Panel Manager Emulator
753 Emulator JSDK Session
754 Event Emulator
755 JVM
801 ネットワーク

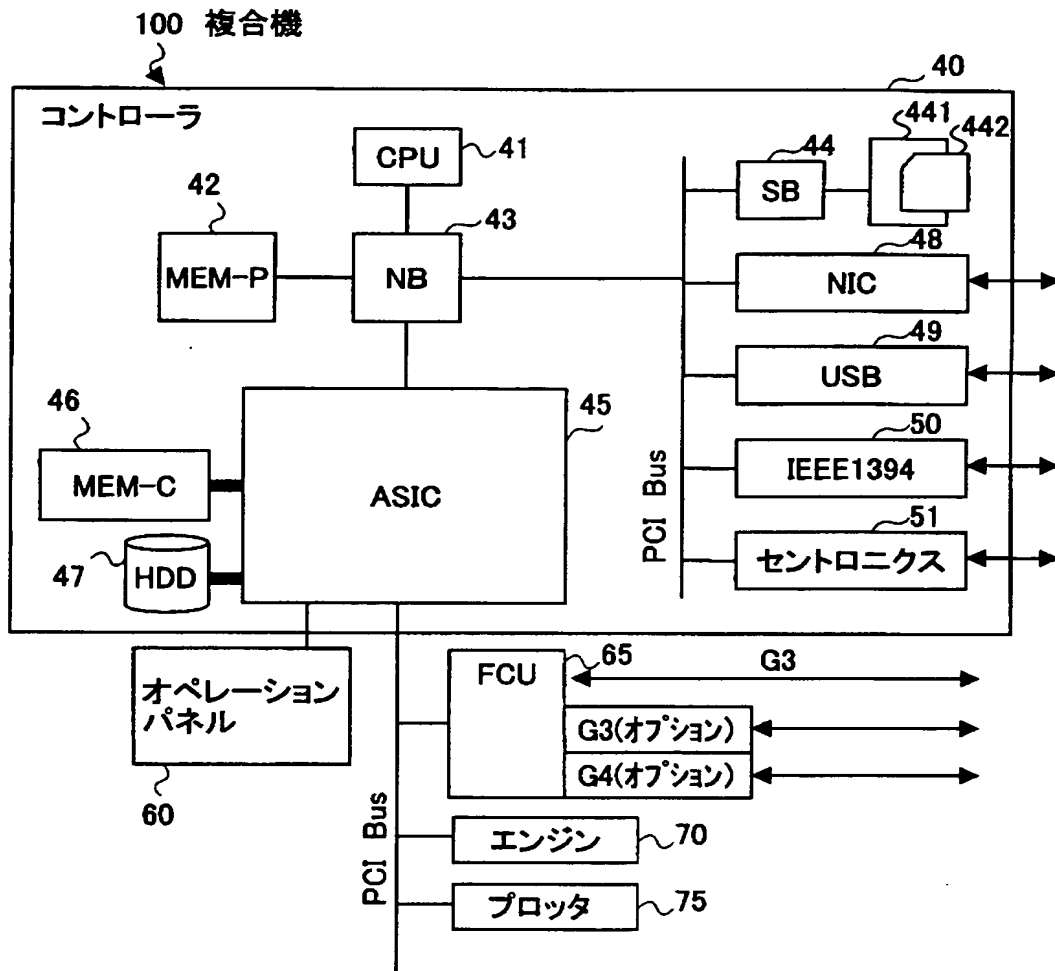
【書類名】 図面
【図 1】

本発明の実施の形態における複合機の構成を示すブロック図



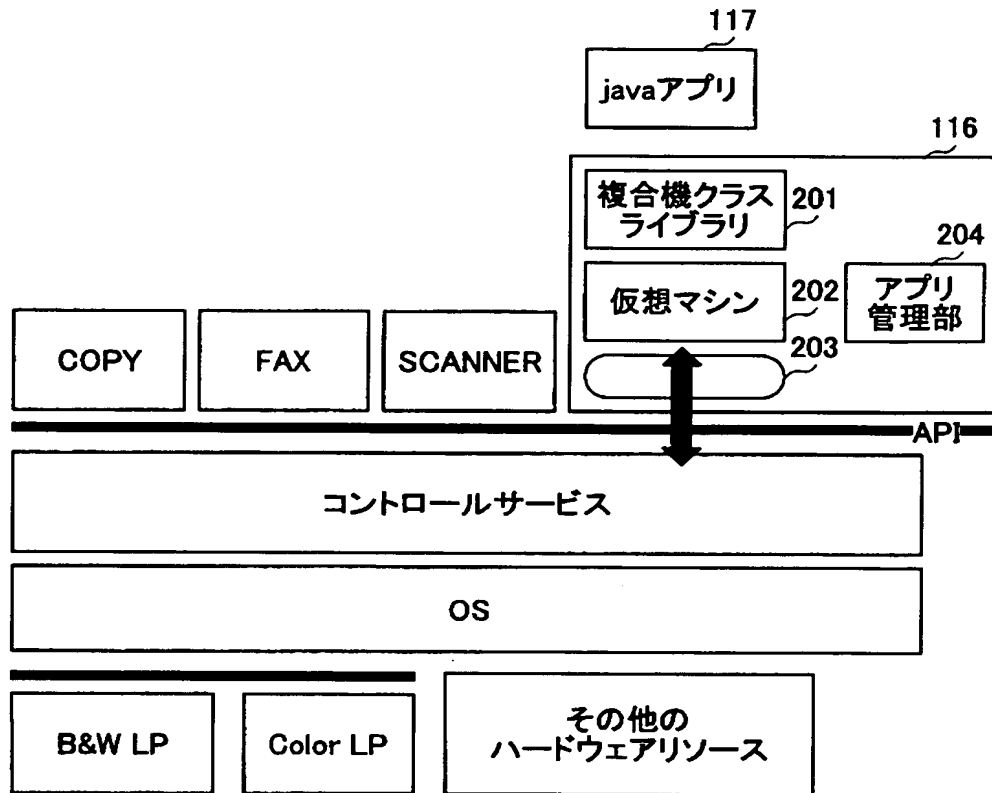
【図 2】

複合機100のハードウェア構成図



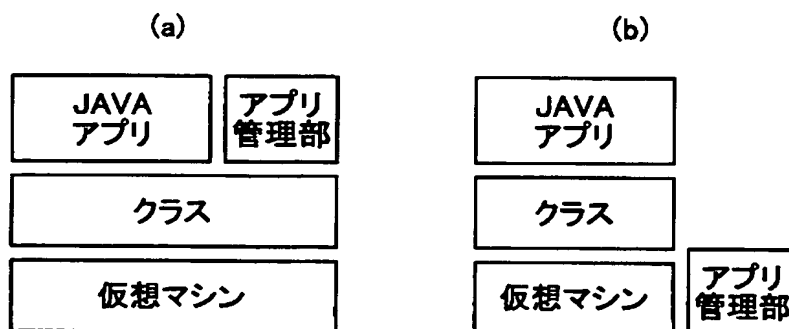
【図 3】

複合機100に搭載されるJava(登録商標)
実行環境116の構成の一例を示す図



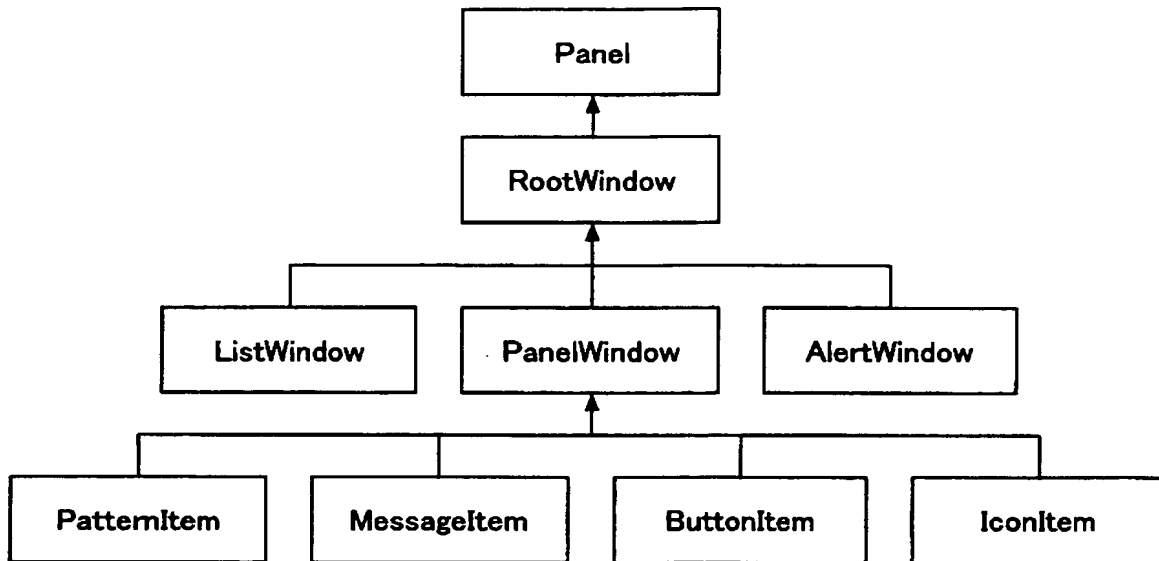
【図 4】

アプリケーション管理部204の実装方法を説明するための図



【図 5】

Java(登録商標)アプリで使用する
操作パネルクラスの階層構成の一例を示す図



【図 6】

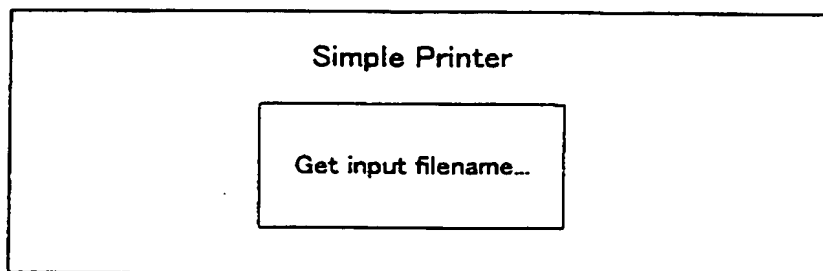
シンプルプリンターのプログラムリスト

```
import java.lang.*;
import com.ricoh.gw.*; //←(1)
public class UserApp extends GWApp //←(2)
{
    public static void main()
    {
        Panel panel = new Panel();
        Printer printer=new Printer(); //←(3)
        PanelWindow mainwindow = new PanelWindow(panel.root());
        ItemString simple = new ItemString("Simple Printer"); //←(4)
        MessageItem salutation = new MessageItem(simple);
        salutation.setRect(100,10,100,10);
        mainwindow.addItem(salutation);
        ItemString inputstr = new ItemString("Get input filename...");
        ButtonItem inputButton = new ButtonItem(); //←(5)
        inputButton.setButtonShape(ButtonItem.BTN_ZAB);
        inputButton.setRect(100,100,100,100);
        inputButton.setWink(true);
        mainwindow.addItem(inputButton);
        inputButton.addChangeListener( //←(6)
            new GWChangeListener()
            {
                public void stateChanged(GWEvent e)
                {
                    String answer =
                        SoftKeyboardWindow.request(
                            "Input filename",
                            "", 256,
                            ItemString.LANG_ENGLISH_US //←(7)
                        );
                    printer.printerDocument( new ItemString(answer) );//←(8)
                }
            }
        )
    }
}
```



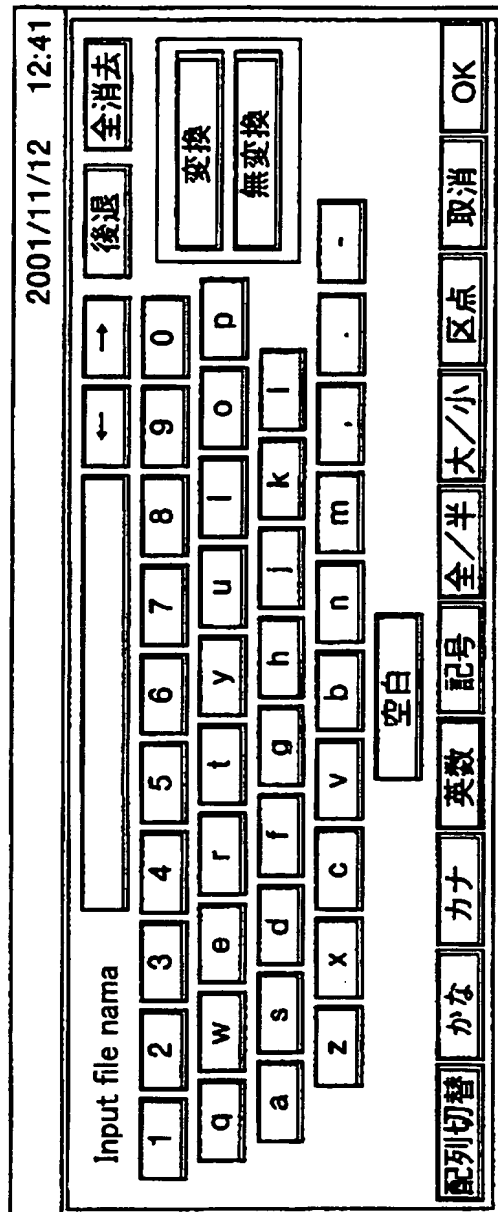
【図 7】

シンプルプリンターにより表示される画面



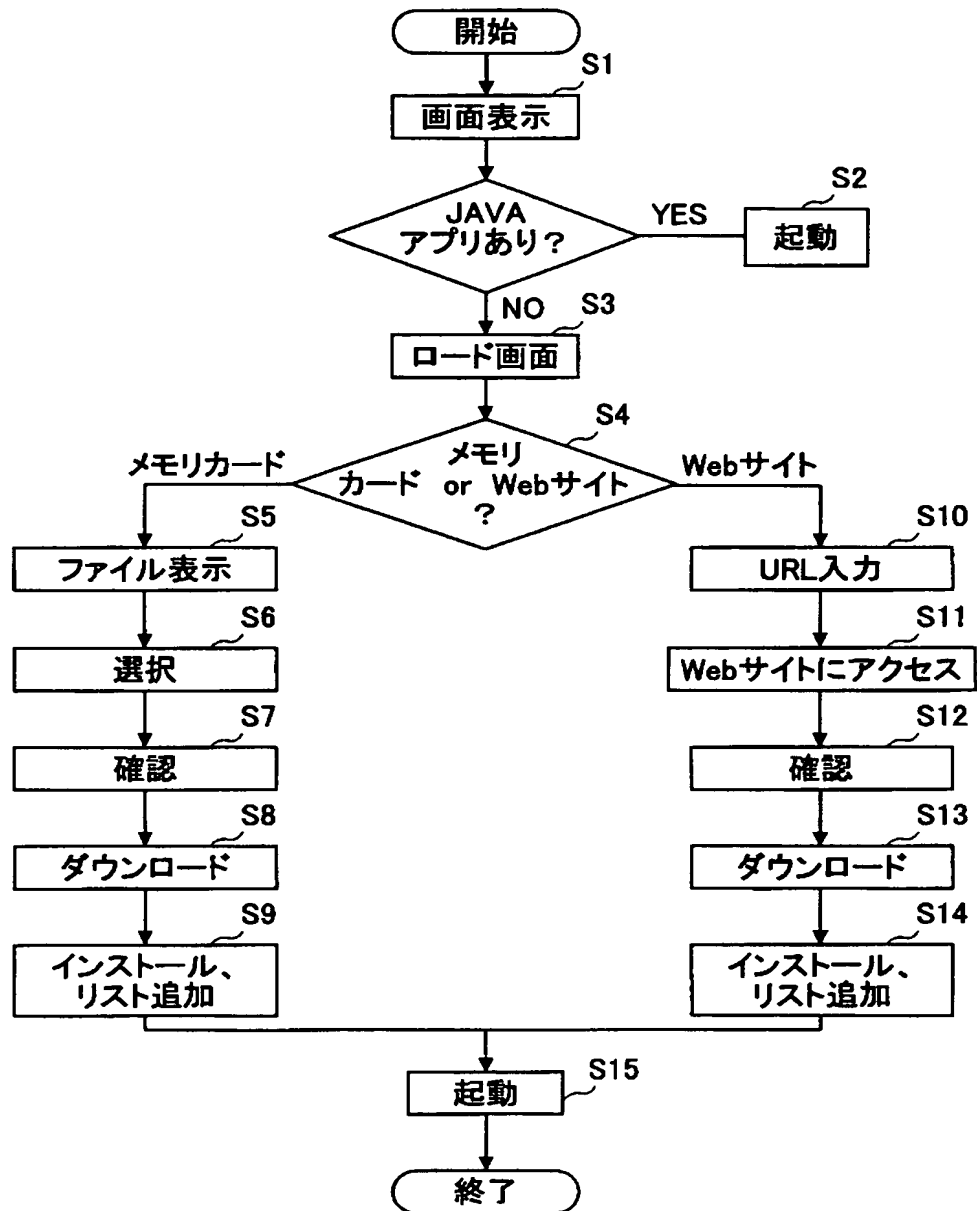
【図 8】

シンプルプリンターにより表示されるソフトキーボードの例を示す図



【図 9】

本発明の第1の実施の形態において
シンプルプリンターを起動するまでの手順を示すフローチャート



【図 10】

本発明の第2の実施の形態におけるJNLPファイルの構文の例を示す図

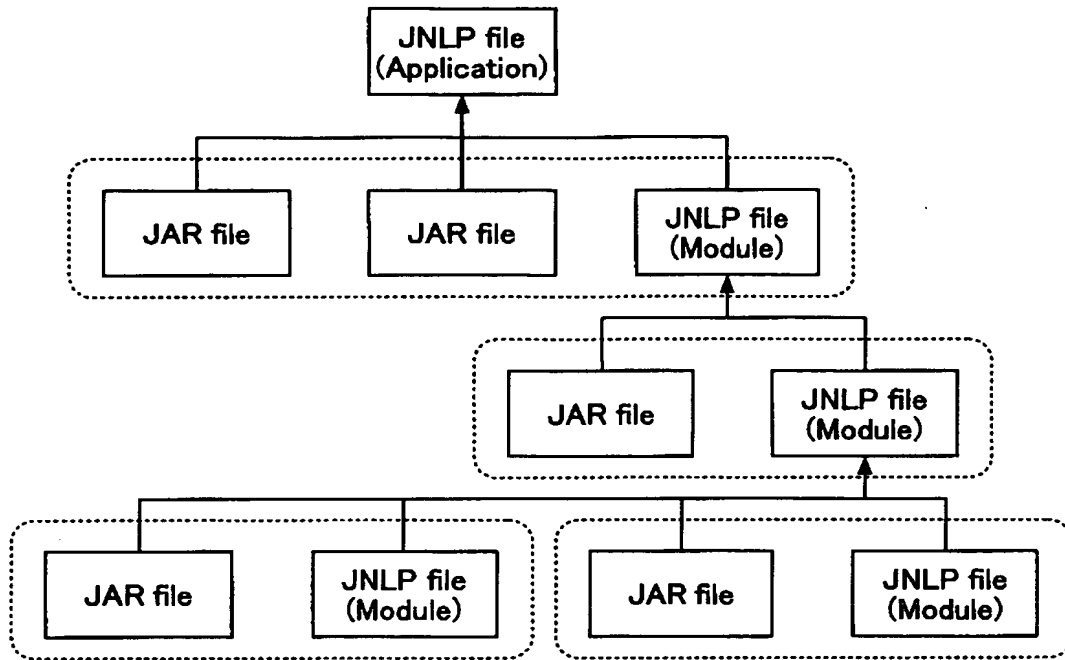
```

<?xml version="1.0" encoding="utf-8"?>
<!-- JNLP File for JSDK Demo Application -->
<jnlp
  spec="1.0+"
  codebase="sdcard://jsdk/0123"
  href="jsdk_demo.jnlp">
  <information>
    <title>Jsdk Demo Application</title>—————1A
    <vendor>JSDK Providers Inc.</vendor>—————1B
    <telephone>092-441-####</telephone>—————1C
    <fax>092-411-####</fax>—————1D
    <product-id>0123456</product-id>—————1E
    <homepage href="docs/help.html"/>
    <description>JSDK Demo Application</description>
    <icon href="images/JSDK_demo.jpg"/>
    <offline-allowed/>
  </information>
  <security>
    <all-permissions/>
  </security>
  <resources>
    <jsdk version="1.0"/>—————3A
    <jar href="JSDKdemo.jar" version="1.0.1"/>—————3B
    <sub-jnlp="http://www.ricoh.co.jp/jsdk/ocr.jnlp"/>————3C
  </resources>
  <update mode="mail">
    <mail address="xxx@yyy.zzz.co.jp"/>
  </update>
  <install mode="auto">
    <destination path="sdcard://jsdk/0123"/>
  </install>
  <xlet-desc class="JSDKdemo" type="noGUI"/>
</jnlp>

```

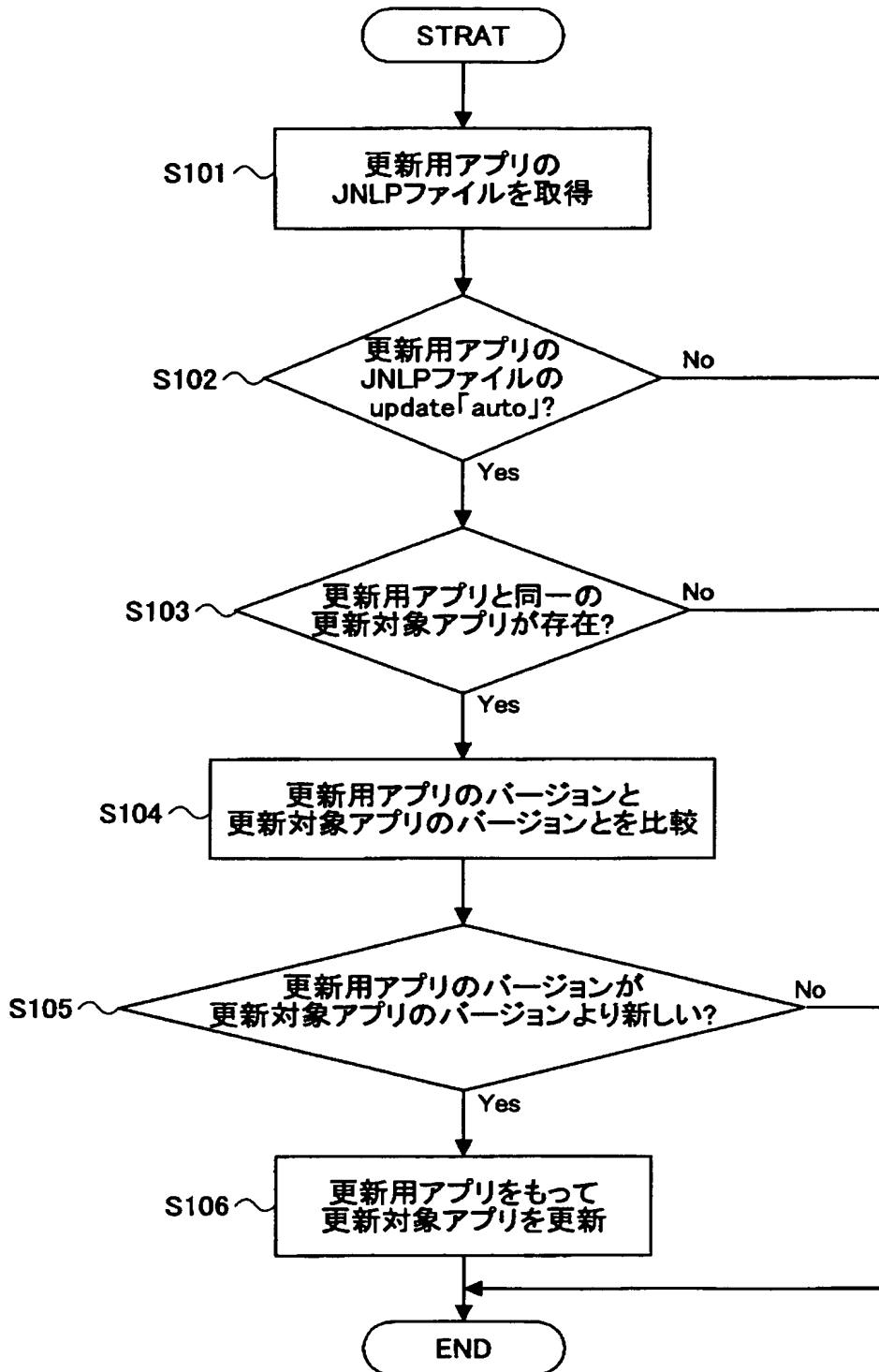
【図 11】

JNLPファイルにおけるツリー構造を示す図

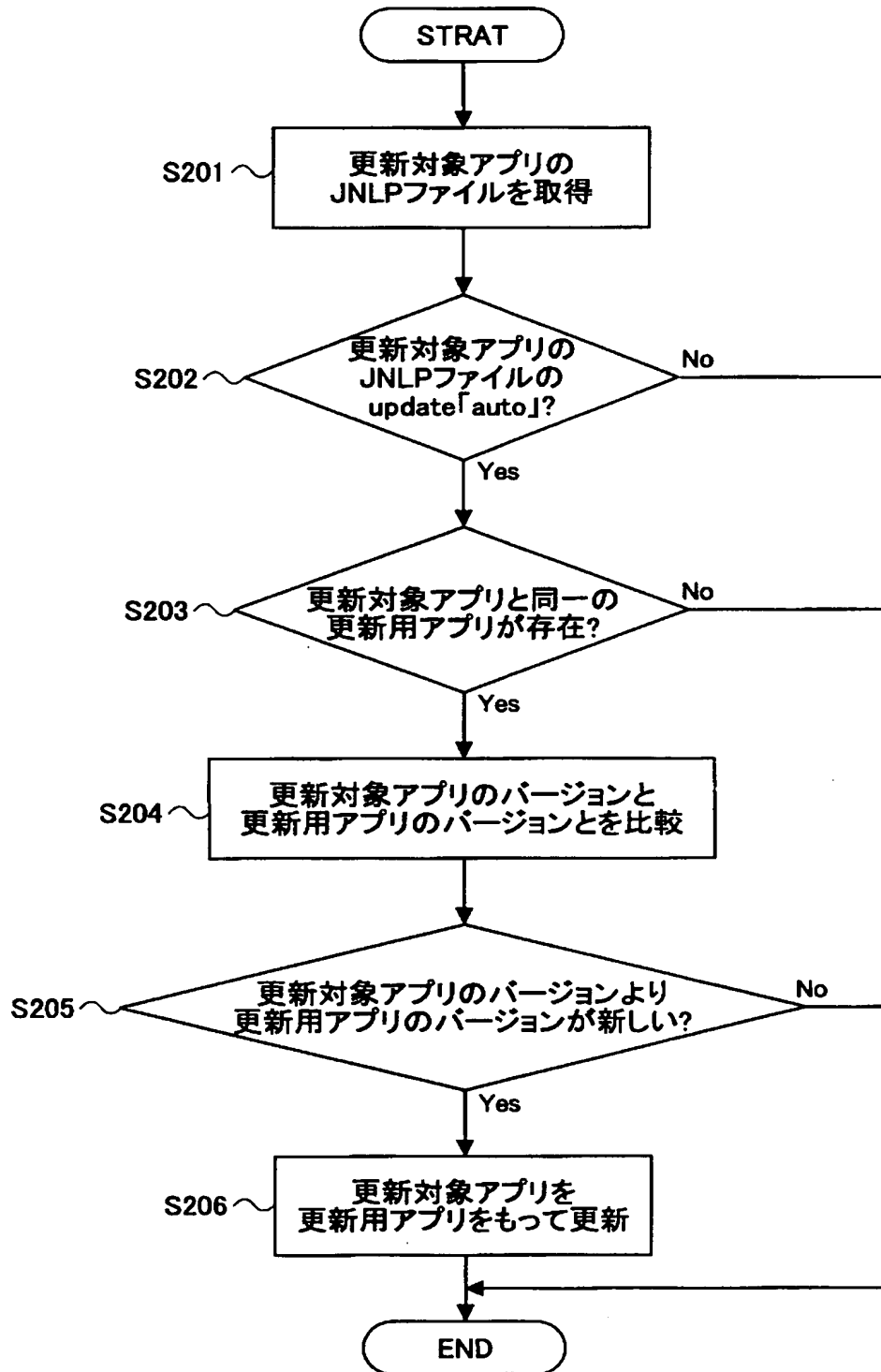


【図 12】

JavaアプリをSDcardからのJSDKアプリに
自動更新する自動更新処理に係るフローチャート

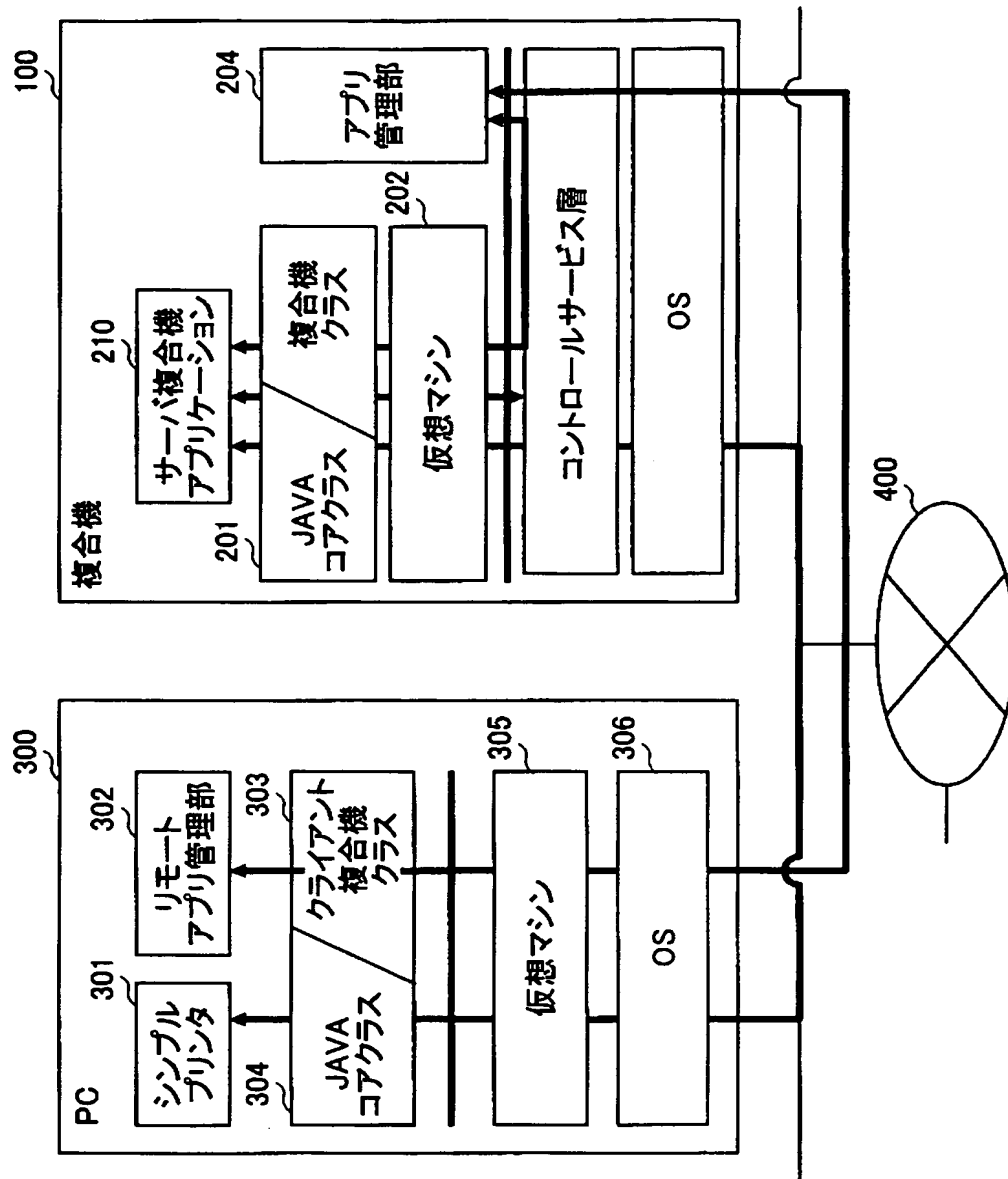


【図 13】

JavaアプリをWebからのJSDKアプリに
自動更新する自動更新処理に係るフローチャート

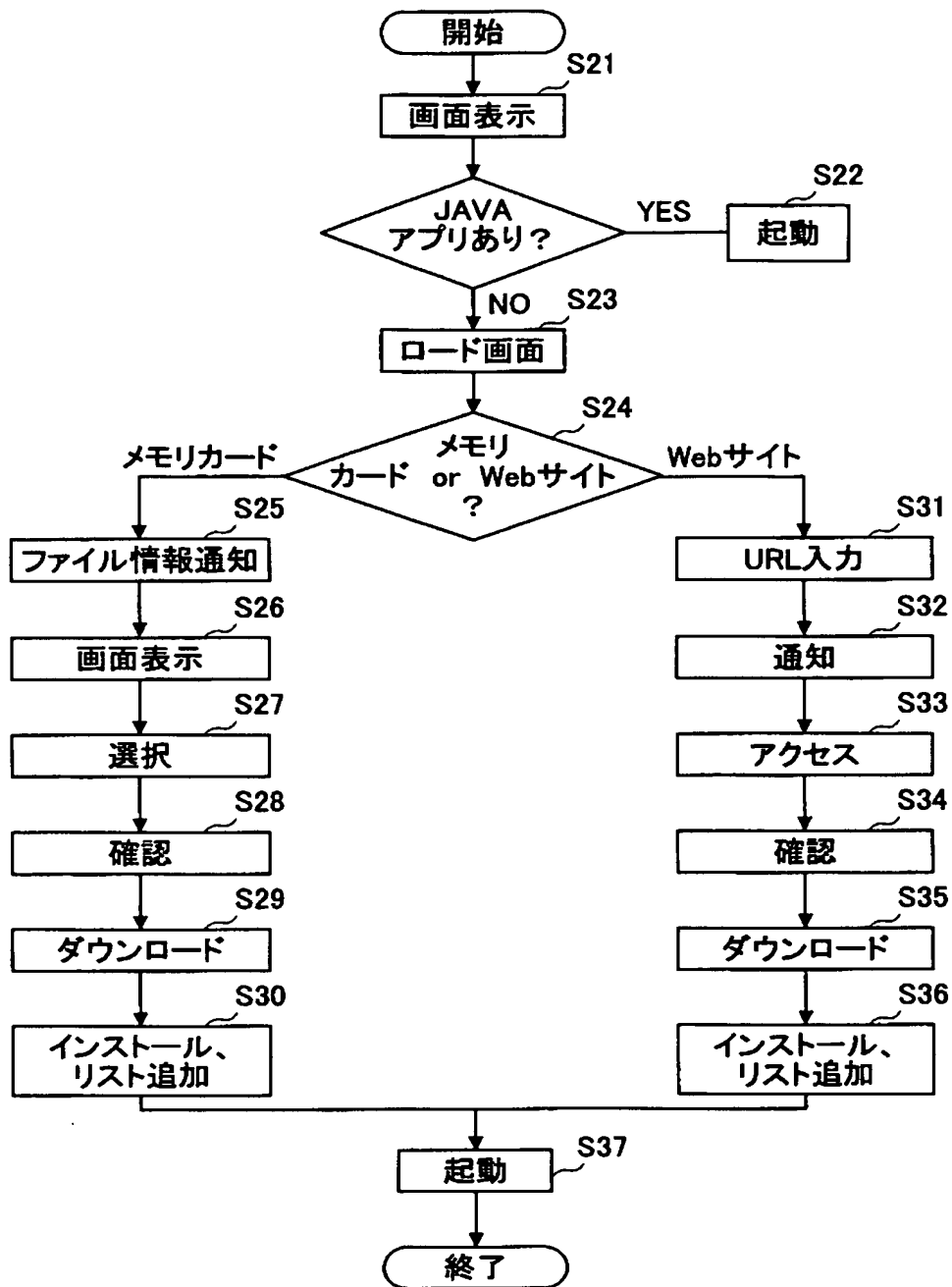
【図 14】

本発明の第3の実施の形態におけるシステム構成図



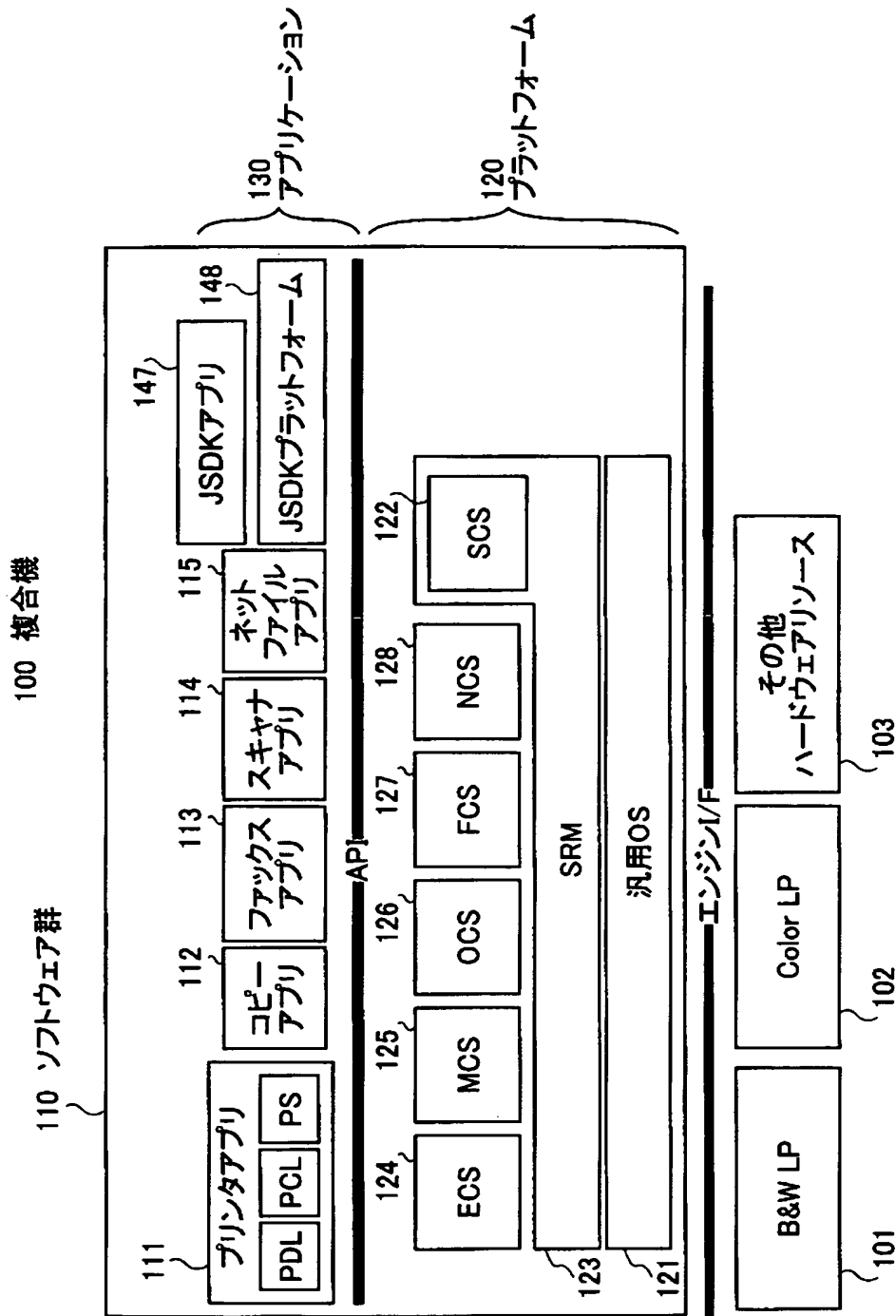
【図 15】

本発明の第3の実施の形態においてシンプルプリンターを
起動するまでの手順を示すフローチャート



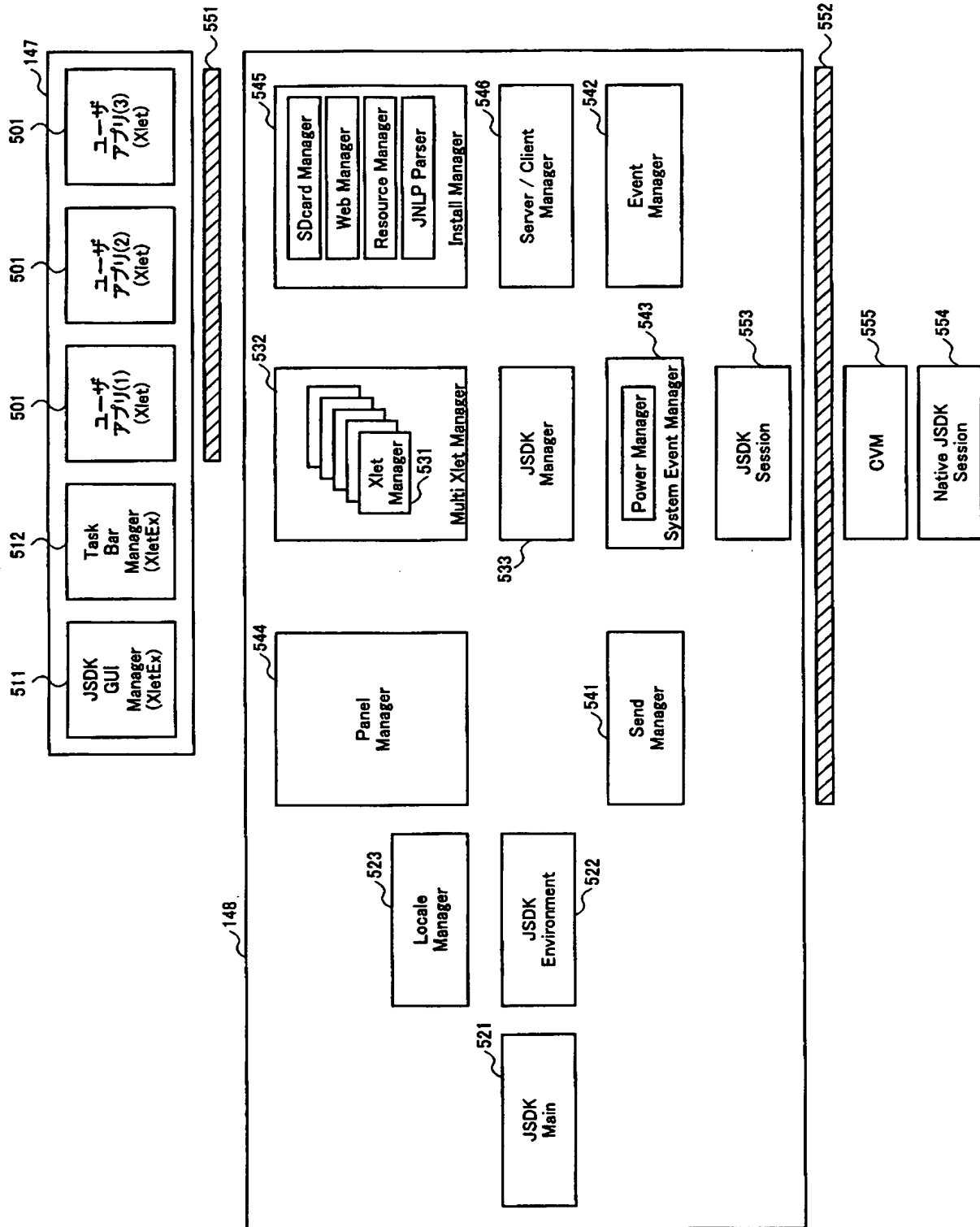
【図 16】

本発明の第4の実施の形態における複合機の構成を示すブロック図



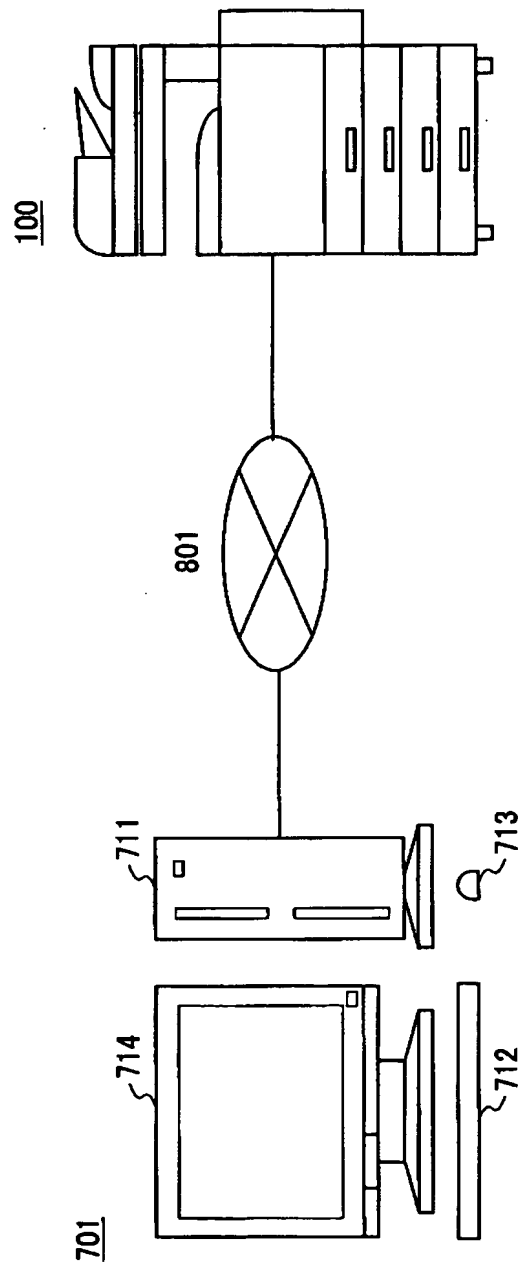
【図 17】

図16のJSDKアプリとJSDKプラットフォームのクラス図



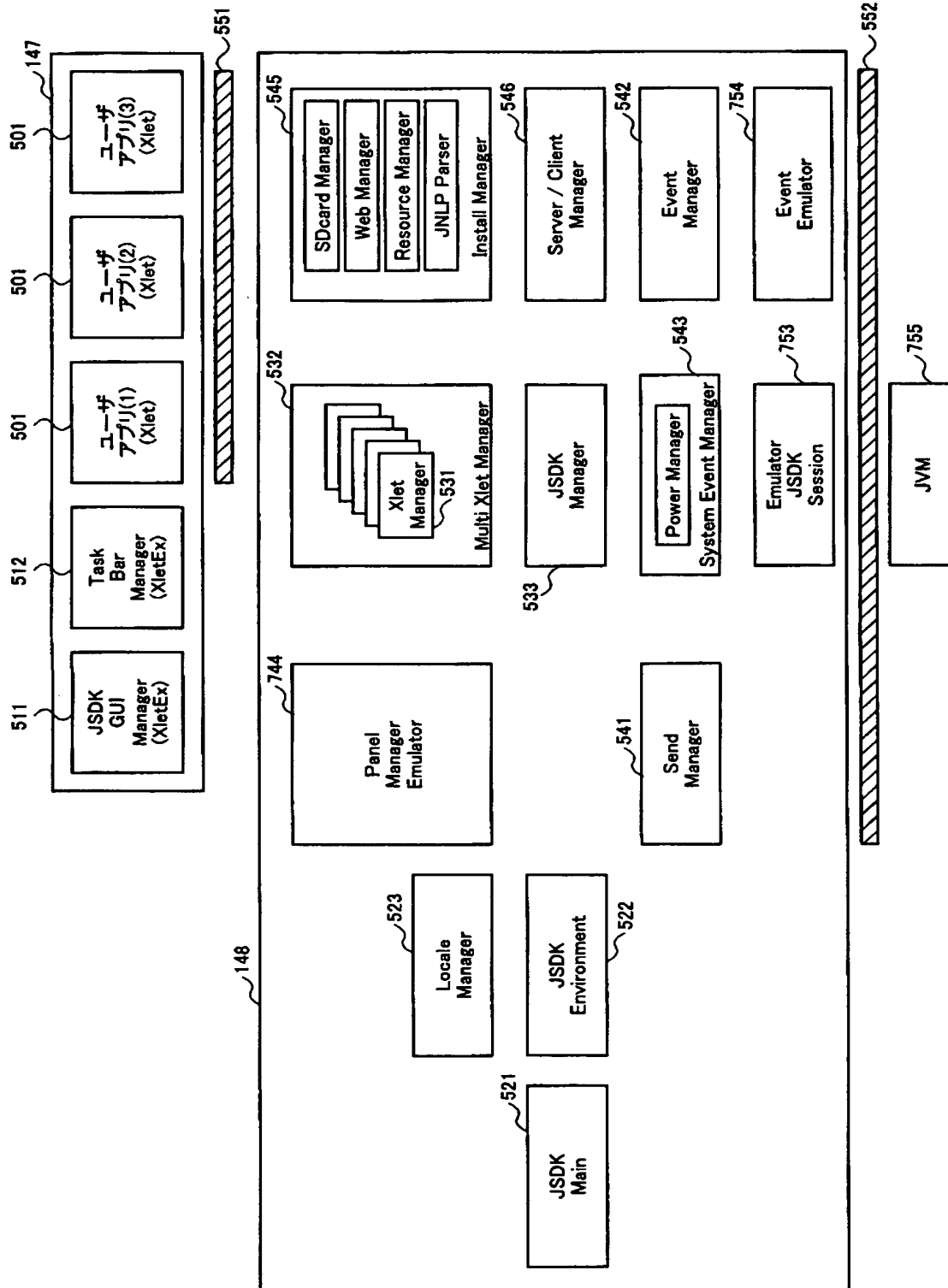
【図 18】

本発明の第4の実施の形態における接続構成図



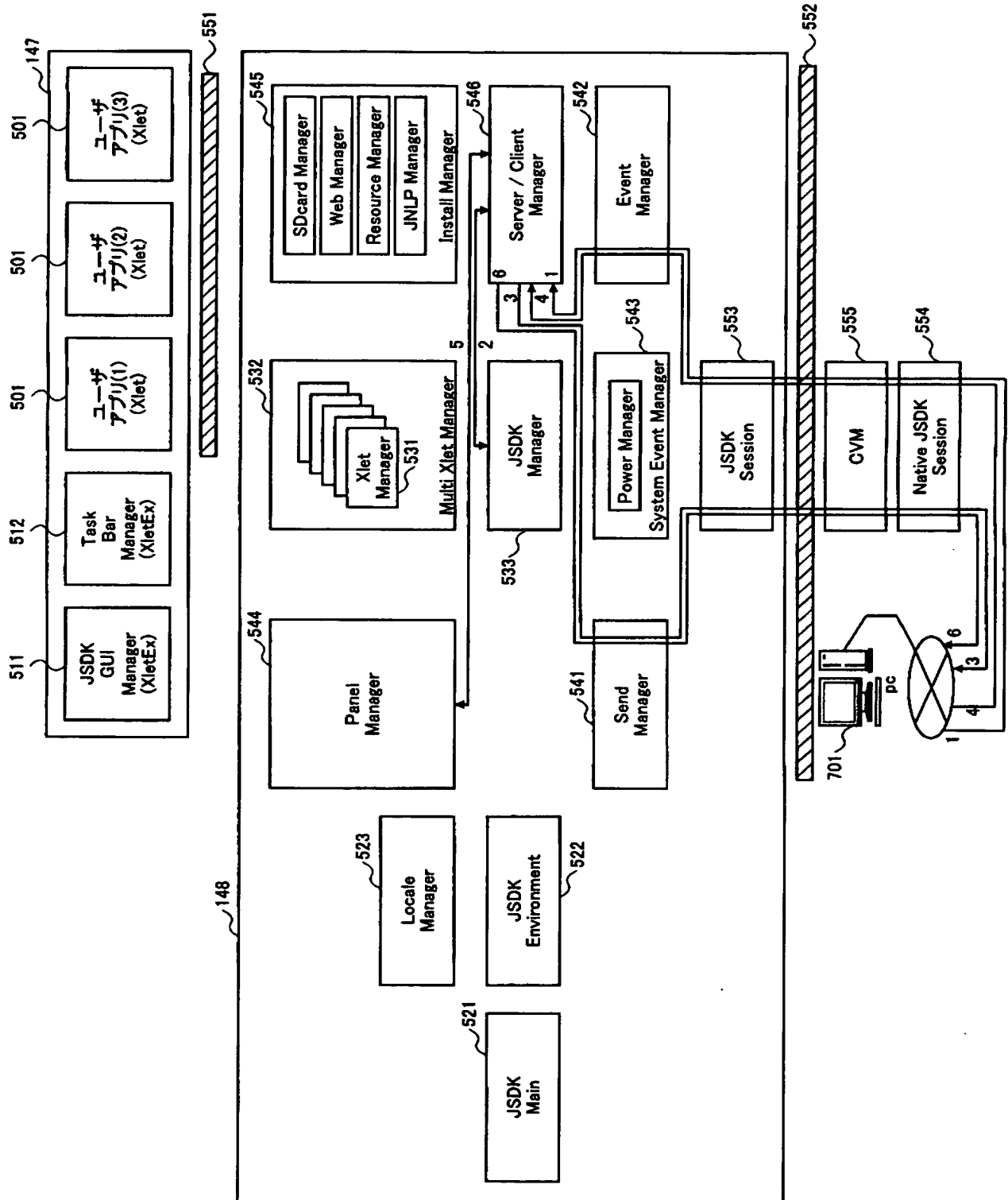
【図 19】

PC内のJSDKアプリとJSDKプラットフォームのクラス図



【図 20】

複合機内で実行される
遠隔操作の準備処理の実行手順について説明するための図



【書類名】 要約書**【要約】**

【課題】 J a v a（登録商標）アプリケーションを複合機上で容易に実行できる環境、及び P C などで実行する J a v a（登録商標）アプリケーションにより複合機を操作できる環境を提供する。

【解決手段】 画像形成処理におけるハードウェア資源の制御に関するサービスを複数のアプリケーションに共通に提供する制御部を備えることにより、複数のアプリケーションを搭載可能に構成された画像形成装置に、前記アプリケーションとしての仮想マシンと、前記仮想マシン上で動作するアプリケーションと、前記仮想マシン上で動作するアプリケーションを管理するためのアプリケーション管理部とを備える。また、ネットワークを介して前記画像形成装置に接続された端末装置からの操作要求を受けて前記画像形成装置を操作するサーバーアプリケーションを備える。

【選択図】 図 3

特願 2004-078979

出願人履歴情報

識別番号

[000006747]

1. 変更年月日

2002年 5月17日

[変更理由]

住所変更

住 所

東京都大田区中馬込1丁目3番6号

氏 名

株式会社リコー